

# Cloud Container Engine

## Guía del usuario

Edición 01  
Fecha 2023-12-05



**Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2024. Todos los derechos reservados.**

Quedan terminantemente prohibidas la reproducción y/o la divulgación totales y/o parciales del presente documento de cualquier forma y/o por cualquier medio sin la previa autorización por escrito de Huawei Cloud Computing Technologies Co., Ltd.

## **Marcas registradas y permisos**



El logotipo  y otras marcas registradas de Huawei pertenecen a Huawei Technologies Co., Ltd. Todas las demás marcas registradas y los otros nombres comerciales mencionados en este documento son propiedad de sus respectivos titulares.

## **Aviso**

Es posible que la totalidad o parte de los productos, las funcionalidades y/o los servicios que figuran en el presente documento no se encuentren dentro del alcance de un contrato vigente entre Huawei Cloud y el cliente. Las funcionalidades, los productos y los servicios adquiridos se limitan a los estipulados en el respectivo contrato. A menos que un contrato especifique lo contrario, ninguna de las afirmaciones, informaciones ni recomendaciones contenidas en el presente documento constituye garantía alguna, ni expresa ni implícita.

Huawei está permanentemente preocupada por la calidad de los contenidos de este documento; sin embargo, ninguna declaración, información ni recomendación aquí contenida constituye garantía alguna, ni expresa ni implícita. La información contenida en este documento se encuentra sujeta a cambios sin previo aviso.

## **Huawei Cloud Computing Technologies Co., Ltd.**

Dirección: Huawei Cloud Data Center Jiaoxinggong Road  
Avenida Qianzhong  
Nuevo distrito de Gui'an  
Gui Zhou, 550029  
República Popular China

Sitio web: <https://www.huaweicloud.com/intl/es-us/>

---

# Índice

---

<b>1 Operaciones y soluciones de alto riesgo.....</b>	<b>1</b>
<b>2 Clústeres.....</b>	<b>7</b>
2.1 Descripción del clúster.....	7
2.1.1 Información básica del clúster.....	7
2.1.2 Clústeres de CCE Turbo y clústeres de CCE.....	10
2.1.3 Comparación de iptables e IPVS.....	13
2.1.4 Notas del lanzamiento de Kubernetes.....	14
2.1.4.1 Notas del lanzamiento de CCE Kubernetes 1.25.....	14
2.1.4.2 Notas del lanzamiento de CCE Kubernetes 1.23.....	16
2.1.4.3 Notas del lanzamiento de CCE Kubernetes 1.21.....	18
2.1.4.4 Kubernetes 1.19 (EOM) Release Notes.....	21
2.1.4.5 Kubernetes 1.17 (EOM) Release Notes.....	23
2.1.4.6 Kubernetes 1.15 (EOM) Release Notes.....	24
2.1.4.7 Kubernetes 1.13 (EOM) Release Notes.....	25
2.1.4.8 Kubernetes 1.11 (EOM) Release Notes.....	26
2.1.4.9 Release Notes for Kubernetes 1.9 (EOM) and Earlier Versions.....	28
2.1.5 Notas de la versión de clúster.....	32
2.2 Compra de un clúster de CCE Turbo.....	48
2.3 Compra de un clúster de CCE.....	52
2.4 Conexión de clústeres.....	56
2.4.1 Conexión a un clúster con kubectl.....	56
2.4.2 Conexión a un clúster con CloudShell.....	60
2.4.3 Conexión a un clúster con un certificado X.509.....	61
2.4.4 Personalización de una SAN de certificados de clúster.....	62
2.4.5 Comandos comunes de kubectl.....	63
2.5 Actualización de un clúster.....	69
2.5.1 Descripción de la actualización.....	69
2.5.2 Antes de comenzar.....	76
2.5.3 Realización de la actualización in situ.....	78
2.5.4 Actualización de sustitución/rodamiento (versión 1.13).....	81
2.5.5 Realización de la verificación posterior a la actualización.....	86
2.5.5.1 Verificación del servicio.....	86
2.5.5.2 Comprobación de pod.....	86

2.5.5.3 Comprobación de red de nodos y contenedores.....	87
2.5.5.4 Comprobación de la etiqueta y la mancha del nodo.....	89
2.5.5.5 Comprobación de nuevo nodo.....	90
2.5.5.6 Comprobación de pod nuevo.....	91
2.5.5.7 Comprobación de salto de nodo para restablecer.....	92
2.5.6 Migración de servicios a través de clústeres de diferentes versiones.....	93
2.5.7 Solución de problemas de excepciones de comprobación previa a la actualización.....	95
2.5.7.1 Realización de la comprobación previa a la actualización.....	95
2.5.7.2 Comprobación del nodo.....	98
2.5.7.3 Comprobación de la lista de bloqueo.....	99
2.5.7.4 Comprobación del complemento.....	100
2.5.7.5 Comprobación del gráfico de Helm.....	101
2.5.7.6 Comprobación de la conectividad SSH del nodo principal.....	102
2.5.7.7 Comprobación del grupo de nodos.....	102
2.5.7.8 Comprobación del grupo de seguridad.....	104
2.5.7.9 Restricción del nodo de Arm.....	105
2.5.7.10 Nodo por migrar.....	106
2.5.7.11 Recurso de Kubernetes descartado.....	106
2.5.7.12 Riesgo de compatibilidad.....	107
2.5.7.13 Versión de nodo de CCEAgent.....	111
2.5.7.14 Uso de la CPU del nodo.....	112
2.5.7.15 Comprobación de CRD.....	113
2.5.7.16 Disco de nodo.....	113
2.5.7.17 Nodo de DNS.....	114
2.5.7.18 Permisos de archivo de directorio de clave de nodo.....	114
2.5.7.19 Kubelet.....	114
2.5.7.20 Memoria de nodos.....	115
2.5.7.21 Servidor de sincronización de reloj de nodo.....	115
2.5.7.22 SO del nodo.....	116
2.5.7.23 Recuento de CPU de nodo.....	116
2.5.7.24 Comando de nodo de Python.....	116
2.5.7.25 Versión de ASM.....	117
2.5.7.26 Preparación del nodo.....	117
2.5.7.27 Diario de nodo.....	118
2.5.7.28 containerd.sock.....	118
2.5.7.29 Error interno.....	118
2.5.7.30 Punto de montaje del nodo.....	119
2.5.7.31 Mancha de nodo de Kubernetes.....	119
2.5.7.32 Restricción de everest.....	120
2.5.7.33 Restricción de cce-hpa-controller.....	121
2.5.7.34 Enlace mejorado del núcleo de la CPU.....	121
2.5.7.35 Estado de componentes de nodo de usuario.....	121

2.5.7.36 Estado de los componentes del nodo del controlador.....	122
2.5.7.37 Límite de recursos de memoria del componente de Kubernetes.....	122
2.5.7.38 API de Kubernetes descartadas.....	122
2.5.7.39 Capacidad de IPv6 de un clúster de CCE Turbo.....	122
2.5.7.40 NetworkManager de nodo.....	123
2.5.7.41 Archivo de ID de nodo.....	123
2.5.7.42 Consistencia de la configuración del nodo.....	124
2.5.7.43 Archivo de configuración de nodo.....	125
2.5.7.44 Consistencia de la configuración de CoreDNS.....	126
2.6 Managing a Cluster.....	128
2.6.1 Gestión de configuración de clúster.....	128
2.6.2 Control de sobrecarga de clúster.....	133
2.6.3 Cambio de escala de clúster.....	134
2.6.4 Eliminación de un clúster.....	136
2.6.5 Renewing a Yearly/Monthly-Billed Cluster.....	139
2.6.6 Hibernación y activación de un clúster (pago por uso).....	140
2.6.7 Changing the Billing Mode from Pay-per-Use to Yearly/Monthly.....	142
2.6.8 Cambio del grupo de seguridad predeterminado de un nodo.....	143
<b>3 Nodos.....</b>	<b>145</b>
3.1 Descripción del nodo.....	145
3.1.1 Precauciones para el uso de un nodo.....	145
3.1.2 Descripción del motor de contenedores.....	147
3.1.3 Descripción del nodo del SO.....	151
3.1.4 Asignación de espacio en disco de datos.....	155
3.1.5 Descripción de los recursos de nodos reservados.....	160
3.1.6 Contenedores de Kata y contenedores comunes.....	163
3.1.7 Número máximo de pods que se pueden crear en un nodo.....	165
3.2 Creación de un nodo.....	166
3.3 Adición de los nodos para gestión.....	175
3.4 Inicio de sesión en un nodo.....	179
3.5 Nodos de gestión.....	180
3.5.1 Gestión de etiquetas de nodo.....	180
3.5.2 Gestión de manchas de nodos.....	182
3.5.3 Restablecimiento de un nodo.....	185
3.5.4 Extracción de un nodo.....	189
3.5.5 Sincronización de datos con servidores en la nube.....	192
3.5.6 Eliminación de un nodo.....	193
3.5.7 Cambio de pago por uso a anual/mensual.....	194
3.5.8 Detención de un nodo.....	194
3.5.9 Realización de actualización de rodamiento para nodos.....	195
3.6 Optimizing Node System Parameters.....	198
3.6.1 Lista de parámetros del sistema de nodos que se pueden optimizar.....	198

3.6.2 Cambio del RuntimeMaxUse de la memoria utilizada por la caché de log en un nodo.....	204
3.6.3 Cambio del número máximo de controladores de archivo.....	205
3.6.4 Modificación de parámetros de núcleo de nodo.....	209
3.6.5 Cambio de los límites de ID de proceso (kernel.pid_max).....	215
3.7 Migración de nodos de Docker a containerd.....	218
<b>4 Grupos de nodos.....</b>	<b>221</b>
4.1 Descripción de pool de nodos.....	221
4.2 Creación de un grupo de nodos.....	225
4.3 Gestión de un grupo de nodos.....	234
4.3.1 Configuración de un grupo de nodos.....	234
4.3.2 Actualización de un grupo de nodos.....	242
4.3.3 Escalamiento de un grupo de nodos.....	249
4.3.4 Copia de un grupo de nodos.....	250
4.3.5 Migración de un nodo.....	251
4.3.6 Eliminación de un pool de nodos.....	252
<b>5 Workloads.....</b>	<b>253</b>
5.1 Overview.....	253
5.2 Creación de una carga de trabajo.....	254
5.2.1 Creación de una Deployment.....	254
5.2.2 Creación de un StatefulSet.....	258
5.2.3 Creación de un DaemonSet.....	263
5.2.4 Creación de un trabajo.....	266
5.2.5 Creación de un trabajo de cron.....	270
5.3 Configuración a Container.....	274
5.3.1 Configuración de información básica del contenedor.....	274
5.3.2 Uso de una imagen de terceros.....	276
5.3.3 Establecimiento de las especificaciones del contenedor.....	278
5.3.4 Setting Container Lifecycle Parameters.....	280
5.3.5 Configuración de la comprobación de estado de un contenedor.....	284
5.3.6 Setting an Environment Variable.....	288
5.3.7 Configuración de la configuración de APM para el análisis de cuello de botella del rendimiento.....	291
5.3.8 Habilitación de reglas de grupo de seguridad ICMP.....	292
5.3.9 Configuración de una política de extracción de imágenes.....	293
5.3.10 Configuración de la sincronización de zona horaria.....	294
5.3.11 Configuración de la política de actualización de carga de trabajo.....	294
5.3.12 Política de programación (afinidad/antiafinidad).....	297
5.3.13 Pod Scale-in Priorities.....	306
5.3.14 Etiquetas y anotaciones de pod.....	307
5.4 Gestión de cargas de trabajo y trabajos.....	310
5.5 Acceso a un contenedor.....	316
<b>6 Planificación.....</b>	<b>319</b>

6.1 Descripción general.....	319
6.2 Programación de CPU.....	321
6.2.1 Política de CPU.....	321
6.2.2 Política de CPU mejorada.....	323
6.3 Programación de GPU.....	325
6.4 Programación de NPU.....	328
6.5 Programación de volcano.....	329
6.5.1 Configuración del volcano como planificador predeterminado.....	330
6.5.2 Despliegue híbrido de trabajos en línea y fuera de línea.....	331
6.5.3 Programación de afinidad de NUMA.....	342
<b>7 Red.....</b>	<b>347</b>
7.1 Descripción general.....	347
7.2 Container Network Models.....	350
7.2.1 Descripción general.....	351
7.2.2 Red de túneles de contenedores.....	353
7.2.3 Red de VPC.....	358
7.2.4 Cloud Native Network 2.0.....	362
7.3 Service.....	372
7.3.1 Descripción general.....	372
7.3.2 ClusterIP.....	375
7.3.3 NodePort.....	379
7.3.4 LoadBalancer.....	383
7.3.4.1 Creación de un LoadBalancer Service.....	383
7.3.4.2 Uso de anotaciones para configurar el balanceo de carga.....	405
7.3.4.3 Service usando HTTP.....	421
7.3.4.4 Configuración de la comprobación de estado para varios puertos.....	423
7.3.4.5 Configuración del estado del pod a través de la comprobación de estado de ELB.....	425
7.3.4.6 Habilitación de redes de paso a través para los servicios de LoadBalancer.....	427
7.3.5 DNAT.....	431
7.3.6 Headless Service.....	436
7.4 Ingresos.....	437
7.4.1 Descripción de entrada.....	437
7.4.2 Ingreso de ELB.....	441
7.4.2.1 Creación de ELB Ingress en la consola.....	441
7.4.2.2 Uso de kubectl para crear una entrada de ELB.....	447
7.4.2.3 Configuración de certificados de HTTPS para ingresos de ELB.....	459
7.4.2.4 Configuración de la Server Name Indication (SNI) para las entradas de ELB.....	464
7.4.2.5 Ingreso de ELB encamina a varios servicios.....	466
7.4.2.6 Ingresos de ELB usando HTTP/2.....	466
7.4.2.7 Configuración de ingresos de ELB con anotaciones.....	468
7.4.3 Ingresos de Nginx.....	475
7.4.3.1 Creación de entradas de Nginx en la consola.....	475

7.4.3.2	Uso de kubectl para crear una entrada de Nginx.....	477
7.4.3.3	Configuración de certificados de HTTPS para entradas de Nginx.....	482
7.4.3.4	Configuración de reglas de reescritura de URL para ingresos de Nginx.....	484
7.4.3.5	Interconexión de ingresos de Nginx con servicios de backend HTTPS.....	487
7.4.3.6	Ingresos de Nginx usando hashing consistente para el balanceo de carga.....	488
7.4.3.7	Configuración de ingresos de Nginx con anotaciones.....	489
7.5	DNS.....	492
7.5.1	Descripción general.....	492
7.5.2	Configuración de DNS.....	494
7.5.3	Uso de CoreDNS para la resolución personalizada de nombres de dominio.....	501
7.5.4	Uso de DNSCache de NodeLocal para mejorar el rendimiento de DNS.....	505
7.6	Configuración de red de contenedores.....	508
7.6.1	Red de host.....	508
7.6.2	Configuración de la limitación de la velocidad de QoS para el acceso entre los pod.....	510
7.6.3	Configuración de la red del túnel del contenedor.....	512
7.6.3.1	Network Policies.....	512
7.6.4	Configuración de Cloud Native Network 2.0.....	515
7.6.4.1	Políticas de grupo de seguridad.....	515
7.7	Configuración de red de clúster.....	518
7.7.1	Adición de un bloque CIDR de VPC secundario para un clúster.....	518
7.7.2	Cambio de una subred de nodo.....	519
7.7.3	Adición de un bloque CIDR de contenedor para un clúster.....	521
7.8	Configuración del acceso dentro de la VPC.....	523
7.9	Acceso a redes públicas desde un contenedor.....	525
<b>8</b>	<b>Almacenamiento de contenedores.....</b>	<b>530</b>
8.1	Conceptos básicos del almacenamiento.....	530
8.2	Descripción del almacenamiento de contenedores.....	533
8.3	Elastic Volume Service (EVS).....	543
8.3.1	Descripción general.....	543
8.3.2	Uso de un disco de EVS existente a través de un PV estático.....	545
8.3.3	Uso de un disco de EVS con un PV dinámico.....	556
8.3.4	Montaje dinámico de un disco de EVS en un StatefulSet.....	563
8.4	Scalable File Service (SFS).....	569
8.4.1	Descripción general.....	569
8.4.2	Uso de un sistema de archivos de SFS existente con un PV estático.....	571
8.4.3	Uso de un sistema de archivos SFS a través de un PV dinámico.....	581
8.4.4	Configuración de las opciones de montaje de volumen de SFS.....	587
8.5	Sistemas de archivos SFS Turbo.....	590
8.5.1	Descripción general.....	590
8.5.2	Uso de un sistema de archivos de SFS Turbo existente con un PV estático.....	591
8.5.3	Configuración de las opciones de montaje de SFS Turbo.....	601
8.5.4	Creación y montaje dinámico de subdirectorios de un sistema de archivos de SFS Turbo.....	603

8.6 Object Storage Service (OBS).....	608
8.6.1 Descripción general.....	608
8.6.2 Uso de un bucket de OBS existente con un PV estático.....	610
8.6.3 Uso de un bucket de OBS con un PV dinámico.....	622
8.6.4 Configuración de opciones de montaje de OBS.....	630
8.6.5 Uso de una AK/SK personalizada para montar un volumen de OBS.....	634
8.6.6 Uso de bucket de OBS en todas las regiones.....	639
8.7 PV local.....	641
8.7.1 Descripción general.....	641
8.7.2 Uso de un PV local a través de un PV dinámico.....	642
8.7.3 Montaje dinámico de un PV local en un StatefulSet.....	647
8.8 Volúmenes efímeros (emptyDir).....	652
8.8.1 Descripción general.....	652
8.8.2 Uso de un EV.....	653
8.9 hostPath.....	658
8.10 StorageClass.....	661
8.11 Grupos de almacenamiento.....	668
8.12 Instantáneas y copias de respaldo.....	670
<b>9 Monitoreo y alarma.....</b>	<b>673</b>
9.1 Resumen de monitoreo.....	673
9.2 Supervisión de métricas personalizadas en AOM.....	676
9.3 Monitoreo de métricas personalizadas con prometheus.....	680
9.4 Monitorización de las métricas del componente del nodo principal.....	687
9.5 Configuraciones de alarma.....	691
<b>10 Logs.....</b>	<b>705</b>
10.1 Descripción general.....	705
10.2 Uso de ICAgent para recopilar logs de contenedores.....	705
<b>11 Namespaces.....</b>	<b>713</b>
11.1 Creación de un espacio de nombres.....	713
11.2 Gestión de espacios de nombres.....	715
11.3 Establecimiento de una cuota de recursos.....	717
<b>12 ConfigMaps y Secretos.....</b>	<b>719</b>
12.1 Creación de un ConfigMap.....	719
12.2 Uso de un ConfigMap.....	721
12.3 Creación de un secreto.....	728
12.4 Uso de un secreto.....	732
12.5 Secretos de clúster.....	739
<b>13 Auto Scaling.....</b>	<b>741</b>
13.1 Descripción general.....	741
13.2 Scaling a Workload.....	743

13.2.1 Mecanismos de ajuste de la carga de trabajo.....	743
13.2.2 Creación de una política de HPA para el escalado automático de cargas de trabajo.....	746
13.2.3 Creación de una política de CustomedHPA para el ajuste automático de cargas de trabajo.....	748
13.2.4 Políticas de CronHPA.....	753
13.2.5 Gestión de políticas de escalado de carga de trabajo.....	760
13.3 Ajuste de un nodo.....	762
13.3.1 Mecanismos de escalado de nodos.....	762
13.3.2 Creación de una política del ajuste de nodos.....	765
13.3.3 Gestión de políticas de escalado de nodos.....	769
13.4 Uso de HPA y CA para el ajuste automático de cargas de trabajo y nodos.....	771
<b>14 Complementos.....</b>	<b>780</b>
14.1 Descripción general.....	780
14.2 coredns (complemento de recursos del sistema, obligatorio).....	784
14.3 everest (complemento de recursos del sistema, obligatorio).....	792
14.4 npd.....	795
14.5 dashboard.....	811
14.6 autoscaler.....	815
14.7 nginx-ingress.....	821
14.8 metrics-server.....	825
14.9 cce-hpa-controller.....	826
14.10 prometheus.....	828
14.11 web-terminal.....	832
14.12 gpu-device-plugin (anteriormente gpu-beta).....	834
14.13 huawei-npu.....	840
14.14 Volcano.....	843
14.15 dew-provider.....	857
14.16 dolphin.....	864
14.17 e-backup.....	868
14.18 node-local-dns.....	879
14.19 kube-prometheus-stack.....	881
14.20 storage-driver (complemento de recursos del sistema, descartado).....	887
<b>15 Gráfico de Helm.....</b>	<b>889</b>
15.1 Descripción general.....	889
15.2 Despliegue de una aplicación desde un gráfico.....	890
15.3 Diferencias entre Helm v2 y Helm v3 y soluciones de adaptación.....	894
15.4 Despliegue de una aplicación a través del cliente de Helm v2.....	896
15.5 Despliegue de una aplicación a través del cliente de Helm v3.....	898
15.6 Convertir una versión de Helm v2 a v3.....	900
<b>16 Permisos.....</b>	<b>903</b>
16.1 Descripción de permisos.....	903
16.2 Permisos de clúster (basados en IAM).....	910

16.3 Permisos de espacio de nombres (basados en Kubernetes RBAC).....	920
16.4 Ejemplo: Diseño y configuración de permisos para usuarios en un departamento.....	928
16.5 Dependencia de permisos de la consola de CCE.....	933
16.6 Seguridad del pod.....	937
16.6.1 Configuración de una política de seguridad de pod.....	937
16.6.2 Configuración de admisión de seguridad de pods.....	940
16.7 Mejora de la seguridad del token de la cuenta de Service.....	943
16.8 Descripción de la confiabilidad del sistema.....	944
<b>17 Cloud Trace Service (CTS).....</b>	<b>947</b>
17.1 Operaciones de CCE con el apoyo de CTS.....	947
17.2 Consulta de logs de CTS.....	952
<b>18 Gestión del almacenamiento: FlexVolume (desusado).....</b>	<b>954</b>
18.1 Descripción de FlexVolume.....	954
18.2 Using EVS Disks as Storage Volumes.....	956
18.2.1 Overview.....	956
18.2.2 (kubectl) Automatically Creating an EVS Disk.....	957
18.2.3 (kubectl) Creación de un PV a partir de un disco de EVS existente.....	958
18.2.4 (kubectl) Creating a Pod Mounted with an EVS Volume.....	967
18.3 Using SFS Turbo File Systems as Storage Volumes.....	970
18.3.1 Overview.....	970
18.3.2 (kubectl) Creating a PV from an Existing SFS Turbo File System.....	971
18.3.3 (kubectl) Creating a Deployment Mounted with an SFS Turbo Volume.....	974
18.3.4 (kubectl) Creating a StatefulSet Mounted with an SFS Turbo Volume.....	975
18.4 Using OBS Buckets as Storage Volumes.....	977
18.4.1 Overview.....	977
18.4.2 (kubectl) Automatically Creating an OBS Volume.....	978
18.4.3 (kubectl) Creación de un PV a partir de un bucket de OBS existente.....	980
18.4.4 (kubectl) Creating a Deployment Mounted with an OBS Volume.....	984
18.4.5 (kubectl) Creating a StatefulSet Mounted with an OBS Volume.....	986
18.5 Using SFS File Systems as Storage Volumes.....	988
18.5.1 Overview.....	988
18.5.2 (kubectl) Automatically Creating an SFS Volume.....	989
18.5.3 (kubectl) Creación de un PV a partir de un sistema de archivos de SFS existente.....	990
18.5.4 (kubectl) Creating a Deployment Mounted with an SFS Volume.....	994
18.5.5 (kubectl) Creating a StatefulSet Mounted with an SFS Volume.....	997

# 1 Operaciones y soluciones de alto riesgo

Durante el despliegue de servicio o la ejecución, puede activar las operaciones de alto riesgo en diferentes niveles, causando fallas de servicio o interrupción. Para ayudarle a estimar mejor y evitar los riesgos de operación, esta sección presenta las consecuencias y soluciones de las operaciones de alto riesgo desde múltiples dimensiones, como clústeres, nodos, redes, balanceo de carga, logs y discos de EVS.

## Clústeres y nodos

**Tabla 1-1** Operaciones y soluciones de alto riesgo

Categoría	Operación	Impacto	Solución
Nodo principal	Modificación del grupo de seguridad de un nodo en un clúster	El nodo principal puede no estar disponible. <b>NOTA</b> Regla de nombre de un nodo principal: <i>Cluster name-<b>cce-control</b>-Random number</i>	Restaura el grupo de seguridad haciendo referencia a <a href="#">Compra de un clúster de CCE</a> y permita que el tráfico del grupo de seguridad pase a través.
	Dejar que el nodo caduque o destruir el nodo	El nodo principal dejará de estar disponible.	Esta operación no se puede deshacer.
	Reinstalación del SO	Se eliminarán los componentes del nodo principal.	Esta operación no se puede deshacer.
	Actualización de componentes en el nodo principal o de etcd	Es posible que el clúster no esté disponible.	Vuelve a la versión original.

Categoría	Operación	Impacto	Solución
	Eliminar o formatear datos del directorio central como <b>/etc/kubernetes</b> en el nodo	El nodo principal dejará de estar disponible.	Esta operación no se puede deshacer.
	Cambiar la dirección IP del nodo	El nodo principal dejará de estar disponible.	Vuelva a cambiar la dirección IP a la original.
	Modificación de los parámetros de los componentes principales (como etcd, kube-apiserver y docker)	El nodo principal puede no estar disponible.	Restaurar la configuración de los parámetros a los valores recomendados. Para obtener más información, véase <a href="#">Gestión de configuración de clúster</a> .
	Sustitución del certificado principal o etcd	Es posible que el clúster no esté disponible.	Esta operación no se puede deshacer.
Nodo de trabajo	Modificación del grupo de seguridad de un nodo en un clúster	Es posible que el nodo no esté disponible. <b>NOTA</b> Regla de nombre de un nodo de trabajo: <i>Cluster name-cce-node-Random number</i>	Restaurar el grupo de seguridad haciendo referencia a <a href="#">Compra de un clúster de CCE</a> y permita que el tráfico del grupo de seguridad pase a través.
	Eliminar el nodo	El nodo dejará de estar disponible.	Esta operación no se puede deshacer.
	Reinstalación del SO	Los componentes del nodo se eliminan y el nodo no está disponible.	Restablezca el nodo. Para obtener más información, véase <a href="#">Restablecimiento de un nodo</a> .

Categoría	Operación	Impacto	Solución
	Actualización del kernel del nodo	El nodo puede no estar disponible o la red puede ser anormal.  <b>NOTA</b> La ejecución del nodo depende de la versión del kernel del sistema. No utilice el comando <b>yum update</b> para actualizar o reinstalar el núcleo del sistema operativo de un nodo a menos que sea necesario. (Reinstalar el kernel del sistema operativo usando la imagen original u otras imágenes es una operación arriesgada.)	Si el sistema operativo es EulerOS 2.2, restaure el nodo o la conectividad de red consultando <b>¿Qué puedo hacer si la red de contenedores no está disponible después de la actualización yum se utiliza para actualizar el sistema operativo?</b>  Si el sistema operativo no es EulerOS 2.2, puede restablecer el nodo. Para más detalles, véase <b>Restablecimiento de un nodo</b> .
	Cambio de la dirección IP del nodo	El nodo dejará de estar disponible.	Vuelva a cambiar la dirección IP a la original.
	Modificación de los parámetros de los componentes principales (como kubelet y kube-proxy)	El nodo puede no estar disponible, y los componentes pueden ser inseguros si se modifican las configuraciones relacionadas con la seguridad.	Restaure la configuración de los parámetros a los valores recomendados. Para obtener más información, véase <b>Configuración de un grupo de nodos</b> .
	Modificación de la configuración del SO	Es posible que el nodo no esté disponible.	Restaure los elementos de configuración o restablezca el nodo. Para obtener más información, véase <b>Restablecimiento de un nodo</b> .
	Eliminar o modificar los directorios <b>/opt/cloud/cce</b> y <b>/var/paas</b> y eliminar el disco de datos	El nodo no estará listo.	Puede restablecer el nodo. Para obtener más información, véase <b>Restablecimiento de un nodo</b> .
	Modificación del permiso de directorio de nodo y del permiso de directorio contenedor	Los permisos serán anormales.	No se recomienda modificar los permisos. Restaure los permisos si se modifican.

Categoría	Operación	Impacto	Solución
	Formatear o particionar discos del sistema, discos de Docker y discos de kubelet en nodos.	Es posible que el nodo no esté disponible.	Restablezca el nodo. Para obtener más información, véase <a href="#">Restablecimiento de un nodo</a> .
	Instalación de otro software en nodos	Esto puede provocar excepciones en los componentes de Kubernetes instalados en el nodo y hacer que el nodo no esté disponible.	Desinstale el software que se ha instalado y restaure o restablezca el nodo. Para obtener más información, véase <a href="#">Restablecimiento de un nodo</a> .
	Modificación de las configuraciones de NetworkManager	El nodo dejará de estar disponible.	Restablezca el nodo. Para obtener más información, véase <a href="#">Restablecimiento de un nodo</a> .
	Eliminar imágenes del sistema como <b>cfe-pause</b> del nodo.	No se pueden crear contenedores y no se pueden extraer imágenes del sistema.	Copie la imagen de otro nodo normal para su restauración.

## Redes y balanceo de carga

Tabla 1-2 Operaciones y soluciones de alto riesgo

Operación	Impacto	Cómo evitar/reparar
Cambiar el valor del parámetro del núcleo <b>net.ipv4.ip_forward</b> a <b>0</b>	La red se vuelve inaccesible.	Cambie el valor a <b>1</b> .
Cambiar el valor del parámetro del núcleo <b>net.ipv4.tcp_tw_recycle</b> a <b>1</b>	El servicio NAT se vuelve anormal.	Cambie el valor a <b>0</b> .
Cambiar el valor del parámetro del núcleo <b>net.ipv4.tcp_tw_reuse</b> a <b>1</b>	La red se vuelve anormal.	Cambie el valor a <b>0</b> .
No configurar el grupo de seguridad de nodo para permitir que los paquetes UDP pasen a través del puerto 53 del bloque CIDR de contenedor	El DNS del clúster no puede funcionar correctamente.	Restaure el grupo de seguridad haciendo referencia a <a href="#">Compra de un clúster de CCE</a> y permita que el tráfico del grupo de seguridad pase a través.

Operación	Impacto	Cómo evitar/reparar
Crear un oyente personalizado en la consola de ELB para el balanceador de carga gestionado por CCE	Los elementos modificados son restablecidos por CCE o la entrada es defectuosa.	Utilizar el archivo YAML del Service para crear automáticamente un oyente.
Vincular un backend definido por el usuario en la consola de ELB al balanceador de carga gestionado por CCE.		No vincular manualmente ningún backend.
Cambiar el certificado de ELB en la consola de ELB para el balanceador de carga gestionado por CCE.		Utilice el archivo YAML de la entrada para gestionar automáticamente los certificados.
Cambiar el nombre de oyente en la consola de ELB para el oyente de ELB gestionado por CCE.		No cambie el nombre del ELB oyente gestionado por CCE.
Cambio de la descripción de balanceadores de carga, oyentes y políticas de reenvío gestionadas por CCE en la consola de ELB.		No modifique la descripción de balanceadores de carga, oyentes o políticas de reenvío gestionadas por CCE.
Elimine los recursos de CRD de las definiciones de network-attachment de default-network.	La red de contenedor está desconectada o el clúster no se puede eliminar.	Si los recursos se eliminan por error, utilice las configuraciones correctas para crear los recursos de red predeterminados.

## Logs

Tabla 1-3 Operaciones y soluciones de alto riesgo

Operación	Impacto	Solución
Eliminar el directorio <b>/tmp/ccs-log-collector/pos</b> de la máquina host	Los logs se recopilan repetidamente.	Ninguna
Eliminar el directorio <b>/tmp/ccs-log-collector/buffer</b> de la máquina host	Los logs se pierden.	Ninguna

## Discos de EVS

**Tabla 1-4** Operaciones y soluciones de alto riesgo

Operación	Impacto	Solución	Notas
Desmontar manualmente de un disco de EVS en la consola	Se notifica un error de E/S cuando se escriben los datos de pod en el disco.	Elimine la ruta de montaje del nodo y vuelva a programar el pod.	El archivo en el pod registra la ubicación donde se van a recopilar los archivos.
Desmontar la ruta de montaje en disco en el nodo	Los datos de pod se escriben en un disco local.	Vuelva a montar la ruta correspondiente al pod.	El búfer contiene archivos de caché de log que se van a consumir.
Funcionar los discos de EVS en el nodo	Los datos de pod se escriben en un disco local.	Ninguna	Ninguna

# 2 Clústeres

---

## 2.1 Descripción del clúster

### 2.1.1 Información básica del clúster

**Kubernetes** es un motor de orquestación de contenedor de código abierto para automatizar el despliegue y la gestión de aplicaciones en contenedores.

Para los desarrolladores, Kubernetes es un sistema operativo de clúster. Kubernetes proporciona detección de servicios, ajuste, equilibrio de carga, autoreparación e incluso elección de líderes, liberando a los desarrolladores de las configuraciones relacionadas con la infraestructura.

Al usar Kubernetes, es como ejecutar un gran número de servidores como uno y el método para implementar aplicaciones en Kubernetes siempre es el mismo.

### Arquitectura de clúster de Kubernetes

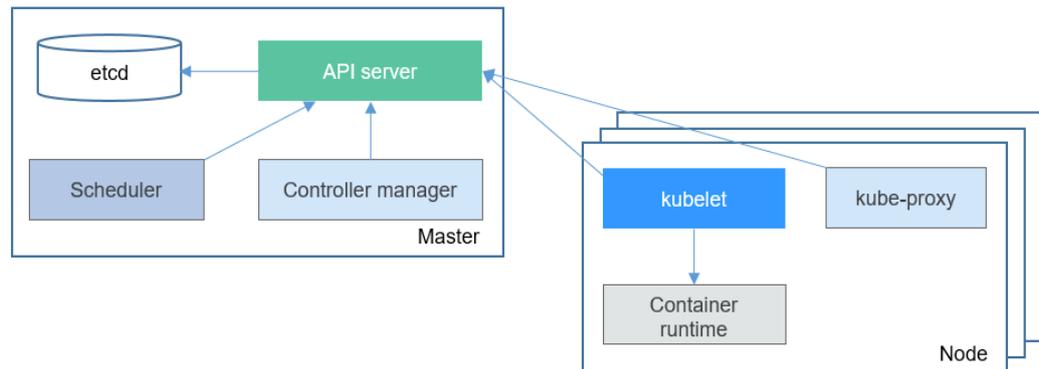
Un clúster de Kubernetes consta de nodos maestros (Masters) y nodos de trabajo (Nodes). Las aplicaciones se despliegan en los nodos de trabajo y puede especificar los nodos para despliegue.

#### NOTA

Para los clústeres de CCE, los nodos maestros están alojados por CCE. Solo necesita crear nodos de trabajo.

La siguiente figura muestra la arquitectura de un clúster de Kubernetes.

**Figura 2-1** Arquitectura de clúster de Kubernetes



### Nodo principal

Un nodo principal es la máquina donde se ejecutan los componentes del plano de control, incluidos el servidor API, Scheduler, Controller manager y etcd.

- Servidor API: funciona como una estación de tránsito para que los componentes se comuniquen entre sí, recibe solicitudes externas y escribe información en etcd.
- Controller manager: realiza funciones a nivel de clúster, como replicación de componentes, seguimiento de nodos y corrección de fallos de nodo.
- Scheduler: programa contenedores a nodos en función de diversas condiciones (como recursos disponibles y afinidad de nodos).
- etcd: sirve como un componente de almacenamiento de datos distribuido que almacena información de configuración del clúster.

En un entorno de producción, se despliegan varios nodos maestros para garantizar una alta disponibilidad del clúster. Por ejemplo, puede desplegar tres nodos maestros para su clúster de CCE.

### Node de trabajo

Un nodo de trabajo es un nodo informático de un clúster, es decir, un nodo que ejecuta aplicaciones en contenedores. Un nodo de trabajo tiene los siguientes componentes:

- kubelet: se comunica con el tiempo de ejecución contenedor, interactúa con el servidor API y gestiona contenedores en el nodo.
- kube-proxy: sirve como proxy de acceso entre los componentes de la aplicación.
- Tiempo de ejecución del contenedor: funciona como el software para ejecutar contenedores. Puede descargar imágenes para crear su tiempo de ejecución de contenedor, como Docker.

## Nodos principales y ajuste de clúster

Cuando crea un clúster en CCE, puede tener uno o tres nodos principales. Tres nodos principales pueden crear un clúster en modo HA.

Las especificaciones del nodo principal deciden el número de nodos que puede gestionar un clúster. Puede seleccionar la escala de gestión de clústeres, por ejemplo, 50 o 200 nodos.

## Red de clústeres

Desde la perspectiva de la red, todos los nodos de un clúster se encuentran en una VPC y contenedores se ejecutan en los nodos. Debe configurar la comunicación nodo-nodo, nodo-contenedor y contenedor-contenedor.

Una red de clúster se puede dividir en tres tipos de red:

- Red de nodos: las direcciones IP se asignan a los nodos de un clúster.
- Red de contenedores: las direcciones IP se asignan a contenedores en un clúster para la comunicación. Actualmente, se soportan múltiples modelos de red de contenedor, y cada modelo tiene su propio mecanismo de trabajo.
- Red de servicios: Un Service es un objeto de Kubernetes que se utiliza para acceder a contenedores. Cada Service tiene una dirección IP fija.

Cuando cree un clúster, seleccione un bloque CIDR adecuado para cada red. Asegúrese de que los bloques CIDR no entren en conflicto entre sí y tengan suficientes direcciones IP disponibles. **No puede cambiar el modelo de red contenedor después de crear el clúster.** Planifique el modelo de red contenedor correctamente con antelación.

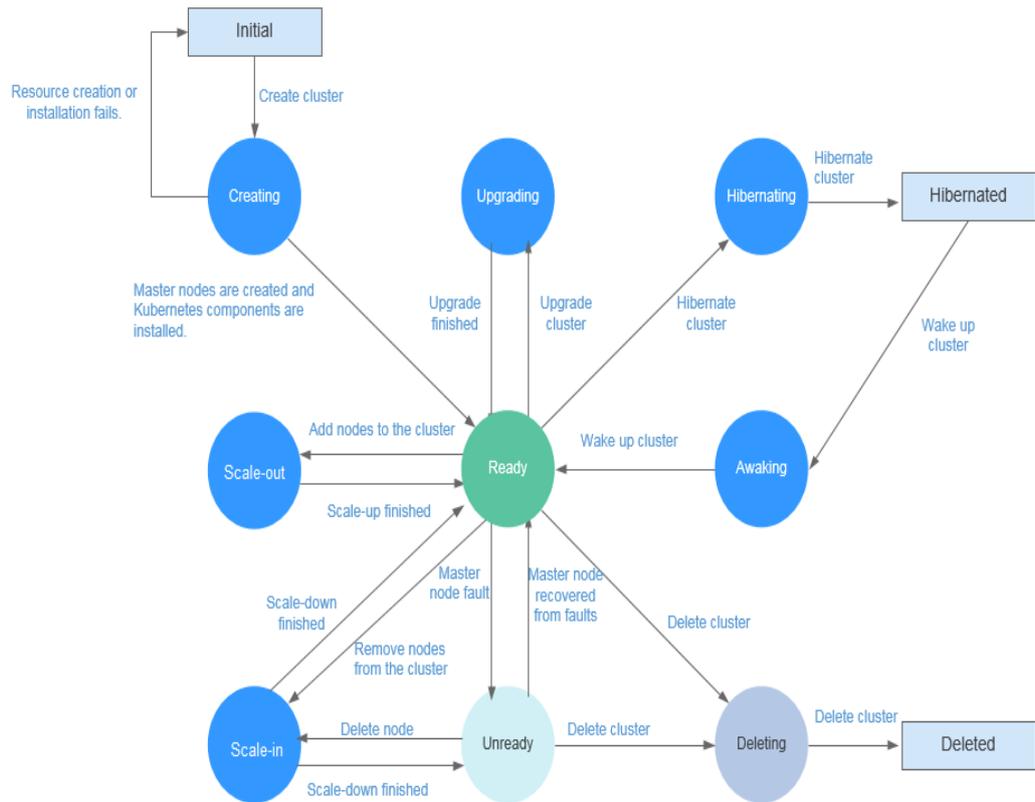
Antes de crear un clúster, se recomienda obtener información sobre la red del clúster y los modelos de red contenedor. Para obtener más información, véase [Container Network Models](#).

## Ciclo de vida del clúster

**Tabla 2-1** Estado del clúster

Estado	Descripción
Creando	Se está creando un clúster y solicita recursos en la nube.
Ejecutando	Un clúster se está ejecutando correctamente.
Expandiendo horizontalmente	Se está agregando un nodo a un clúster.
Reduciendo horizontalmente	Se está eliminando un nodo de un clúster.
Hibernando	Un clúster está hibernando.
Despertando	Se está despertando un clúster.
Actualizando	Se está actualizando un clúster.
No disponible	Un clúster no está disponible.
Eliminando	Se está eliminando un clúster.

**Figura 2-2** Transición de estado de clúster



## 2.1.2 Clústeres de CCE Turbo y clústeres de CCE

### Comparación entre clústeres de Turbo CCE y de CCE

En la siguiente tabla se enumeran las diferencias entre los clústeres de Turbo CCE y los clústeres de CCE:

**Tabla 2-2** Tipos de clúster

Dimensión	Subdimensión	Clúster de Turbo de CCE	Clúster de CCE
Clúster	Posicionamiento	Clúster de contenedores de próxima generación para Cloud Native 2.0 con cómputo, redes y programación acelerados	Clúster estándar para el uso comercial común
	Tipo de nodo	Implementación híbrida de máquinas virtuales y servidores bare-metal	Implementación híbrida de máquinas virtuales y servidores bare-metal

Dimensión	Subdimensión	Clúster de Turbo de CCE	Clúster de CCE
Redes	Modelo	<b>Cloud Native Network 2.0:</b> se aplica a escenarios de gran escala y alto rendimiento. Escala de red: 2000 nodos	<b>Cloud-native network 1.0</b> para escenarios que no requieren un alto rendimiento o implican una implementación a gran escala. <ul style="list-style-type: none"> <li>● Modelo de red de túneles</li> <li>● Modelo de red de VPC</li> </ul>
	Rendimiento	La red de la VPC y la red de contenedores se aplanan en una sola, logrando una pérdida de rendimiento nula.	La red de VPC se superpone con la red de contenedores, causando cierta pérdida de rendimiento.
	Aislamiento de la red de contenedores	Los pods se pueden asociar directamente con grupos de seguridad para configurar políticas de aislamiento para recursos dentro y fuera de un clúster.	<ul style="list-style-type: none"> <li>● Modelo de red de túnel: las políticas de aislamiento de red se admiten para la comunicación entre clústeres (mediante la configuración de políticas de red).</li> <li>● Modelo de red de VPC: no se admite el aislamiento.</li> </ul>
Seguridad	Aislamiento	<ul style="list-style-type: none"> <li>● Máquina física: contenedores Kata, proporcionando aislamiento a nivel de VM.</li> <li>● VM: Se implementan los contenedores comunes.</li> </ul>	Los contenedores comunes son desplegados y aislados por Cgroups.

## Descripción del rendimiento en la creación de pods por lotes en un clúster de CCE Turbo

Los pods de un clúster de CCE Turbo solicitan las interfaces de red elástica (ENI) o sub-ENI de la VPC. Actualmente, los pods se vinculan con ENIs o sub-ENIs una vez completada la programación de pods. La velocidad de creación de pod está limitada por la rapidez con la que se crean y enlazan las NIC. En la siguiente tabla se describen las restricciones.

**Tabla 2-3** Duración de la creación de ENI

Tipo de nodo	Tipo de ENI	Número máximo de ENI admitidas	Vinculación de ENI al nodo	Disponibilidad de ENI	Control de simultaneidad	Configuración de previnculación
ECS	Sub-ENI	256	Especifique la ENI del nodo para crear una sub-ENI.	En un segundo	Nivel de tenant : 600/minuto	<p>Para los clústeres anteriores a 1.19.16-r2, 1.21.5-r0 y 1.23.3-r0, no se admite la vinculación previa.</p> <p>Para los clústeres de 1.19.16-r2, 1.21.5-r0, 1.23.3-r0 a 1.19.16-r4, 1.21.7-r0 y 1.23.5-r0, se soporta la previnculación dinámica (nic-minimum-target=10; nic-warm-target=2).</p> <p>Para los clústeres de 1.19.16-r4, 1.21.7-r0, 1.23.5-r0, 1.25.1-r0 y posteriores, se admite la previnculación dinámica (nic-minimum-target=10; nic-maximum-target=2; nic-warm-target=2; nic-max-above-warm-target=2).</p>
BMS	ENI	128	Vincula una ENI a un nodo.	20s-30s	Nivel de nodo: 3 simultáneamente	<p>Para los clústeres anteriores a 1.19.16-r4, 1.21.7-r0 y 1.23.5-r0, el número total de las ENI se basa en la relación umbral (nic-threshold=0.3:0.6).</p> <p>Para los clústeres de 1.19.16-r4, 1.21.7-r0, 1.23.5-r0, 1.25.1-r0, y posteriores, se admite la previnculación dinámica (nic-minimum-target=10; nic-maximum-target=2; nic-warm-target=2; nic-max-above-warm-target=2).</p>

**Creación de pods en los nodos de ECS (con las sub-ENI)**

- Si no hay ninguna ENI previnculada disponible en el nodo al que está programado el pod, se invoca a la API de crear una sub-ENI para crear una sub-ENI en una ENI del nodo y asignar la sub-ENI al pod.
- Si una ENI previnculada está disponible en el nodo para el que está programado el pod, la primera sub-ENI no utilizada se asigna al pod.

- Limitado por la velocidad de creación simultánea de las sub-ENI, se puede crear un máximo de 600 pods por minuto sin vinculación previa. Si se requiere una creación a mayor escala, puede configurar la vinculación previa para las sub-ENI.

#### **Creación de pods en los nodos de BMS (con las sub-ENI)**

- Si no hay ninguna ENI prelimitada disponible en el nodo al que está programado el pod, se invoca a la API para vincular una ENI al nodo para vincular y asignar una ENI al pod. Se tarda aproximadamente de 20 a 30 segundos en unir una ENI a un nodo de BMS.
- Si una ENI previnculada está disponible en el nodo para el que está programado el pod, la primera ENI no utilizada se asigna al pod.
- Limitada por la velocidad de enlace de las ENI a los nodos de BMS, la velocidad de inicio de los pods en el mismo nodo es de 3/20 segundos sin la vinculación previa. Por lo tanto, se recomienda previncular las ENI para los nodos de BMS.

### **2.1.3 Comparación de iptables e IPVS**

kube-proxy es un componente clave de un clúster de Kubernetes. Es responsable del balanceo de carga y el reenvío entre un Service y su pod de backend.

CCE admite dos modos de reenvío: iptables e IPVS.

- IPVS permite un mayor rendimiento y un reenvío más rápido. Este modo se aplica a escenarios en los que el ajuste del clúster es grande o el número de servicios es grande.
- iptables es el modo kube-proxy tradicional. Este modo se aplica al escenario donde el número de servicios es pequeño o un gran número de conexiones cortas se envían simultáneamente en el cliente.

#### **Restricciones**

En un clúster que usa el modo proxy IPVS, si el ingreso y el Service usan el mismo balanceador de carga de ELB, no se puede acceder a la entrada desde los nodos y contenedores en el clúster porque kube-proxy monta la dirección de LoadBalancer Service en el puente ipvs-0. Este puente intercepta el tráfico del balanceador de carga conectado a la entrada. Se recomienda utilizar diferentes balanceadores de carga de ELB para la entrada y el Service.

#### **iptables**

iptables es una función del kernel de Linux que proporciona una gran cantidad de capacidades de procesamiento y filtrado de paquetes de datos. Permite unir secuencias flexibles de reglas a varios ganchos en la canalización de procesamiento de paquetes. Cuando se usa iptables, kube-proxy implementa NAT y balanceo de carga en el gancho de pre-routing de NAT.

kube-proxy es un algoritmo  $O(n)$ , en el que  $n$  aumenta con el ajuste de clúster. El ajuste del clúster se refiere al número de servicios y pods de backend.

#### **IPVS**

IP Virtual Server (IPVS) está construido sobre Netfilter e implementa el balanceo de carga de la capa de transporte como parte del núcleo de Linux. IPVS puede dirigir solicitudes de servicios basados en TCP y UDP a los servidores reales, y hacer que los servicios de los servidores reales aparezcan como servicios virtuales en una sola dirección IP.

En el modo IPVS, kube-proxy utiliza balanceo de carga de IPVS en lugar de iptables. IPVS está diseñado para equilibrar cargas para un gran número de Services. Tiene un conjunto de API optimizadas y utiliza algoritmos de búsqueda optimizados en lugar de simplemente buscar reglas de una lista.

La complejidad del proceso de conexión del kube-proxy basado en IPVS es O(1). En otras palabras, en la mayoría de los casos, la eficiencia del procesamiento de la conexión es irrelevante para la escala de clúster.

IPVS implica múltiples algoritmos de balanceo de carga, tales como round-robin, el retardo esperado más corto, las conexiones mínimas y varios métodos de hash. Sin embargo, iptables solo tiene un algoritmo para la selección aleatoria.

En comparación con iptables, IPVS tiene las siguientes ventajas:

1. Proporciona una mejor escalabilidad y rendimiento para los clústeres grandes.
2. Soporta mejores algoritmos de balanceo de carga que iptables.
3. Admite funciones que incluyen comprobación del estado del servidor y reintentos de conexión.

## 2.1.4 Notas del lanzamiento de Kubernetes

### 2.1.4.1 Notas del lanzamiento de CCE Kubernetes 1.25

CCE ha aprobado el Certified Kubernetes Conformance Program y es una oferta certificada de Kubernetes. En esta sección se describen las actualizaciones de CCE Kubernetes 1.25.

#### Actualizaciones de versiones

CCE proporciona optimización y actualización de componentes de enlace completo para Kubernetes 1.25.

**Tabla 2-4** Componentes principales y descripciones

Tipo de clúster	componente del núcleo	Versión	Precauciones
Clúster de CCE	Kubernetes	1.25	Ninguna
	Docker	EulerOS/CentOS: 18.9.0 Ubuntu: 18.09.9	Ninguna
	containerd	1.4.1	Ninguna
	OS	CentOS Linux release 7.6	Ninguna
		EulerOS release 2.9	Ninguna
		EulerOS release 2.5	Ninguna
		Ubuntu 18.04 server 64-bit	Ninguna

Tipo de clúster	componente del núcleo	Versión	Precauciones
		Huawei Cloud EulerOS 2.0	Ninguna
Clúster de Turbo de CCE	Kubernetes	1.25	Ninguna
	Docker	EulerOS/CentOS: 18.9.0 Ubuntu: 18.09.9	Ninguna
	containerd	1.4.1	Ninguna
	OS (ECS)	CentOS Linux release 7.6 Ubuntu 18.04 server 64-bit EulerOS release 2.9	Ninguna
	OS (BMS)	CentOS Linux release 7.6 Ubuntu 18.04 server 64-bit EulerOS release 2.9	Ninguna

## Cambios y depreciaciones de recursos

### Cambios en CCE 1.25

Excepto EulerOS 2.5, todos los nodos de CCE en el clúster 1.25 utilizan containerd de forma predeterminada.

### Notas del lanzamiento de Kubernetes 1.25

- PodSecurityPolicy se sustituye por Pod Security Admission. Para obtener más información sobre la migración, consulte [Migrar desde PodSecurityPolicy al controlador de admisión integrado de PodSecurity](#).
- SeccompDefault está en Beta. Para habilitar esta función, debe agregar el parámetro de inicio `--seccomp-default=true` a kubelet. De esta manera, seccomp se establece en **RuntimeDefault** de forma predeterminada, mejorando la seguridad del sistema. Los clústeres de v1.25 ya no usan `seccomp.security.alpha.kubernetes.io/pod` y `contenedor.seccomp.security.alpha.kubernetes.io/annotation` para usar seccomp. Reemplácelos por el campo `securityContext.seccompProfile` en el pod o el contenedor. Para obtener más información, consulte [Configurar un contexto de seguridad para un pod o un contenedor](#).

#### NOTA

Una vez habilitada la función, las llamadas al sistema requeridas por la aplicación pueden estar restringidas por el tiempo de ejecución. Asegúrese de que la depuración se realiza en el entorno de prueba y la aplicación no se ve afectada.

- EndPort en la política de red es estable. Esta característica se incorpora en la versión 1.21. EndPort se agrega a NetworkPolicy para que pueda especificar un rango de puertos.
- A partir de la v1.25, Kubernetes ya no admite la autenticación de certificados generada mediante el algoritmo SHA1WithRSA o ECDSAWithSHA1. Se recomienda utilizar el algoritmo SHA256.

#### Notas de lanzamiento de Kubernetes 1.24

- Dockershim fue marcado como obsoleto en Kubernetes 1.20 y oficialmente eliminado del código kubelet en Kubernetes 1.24. Si desea utilizar contenedor Docker, cambie a cri-dockerd u otros tiempos de ejecución que admitan CRI, como containerd y CRI-O. Para obtener más información sobre cómo cambiar de motor Docker a containerd, consulte [Migración de nodos de Docker a containerd](#).

#### NOTA

Compruebe si hay agentes o aplicaciones que dependen de Docker. Por ejemplo, si se utilizan **docker ps**, **docker run** y **docker inspect** asegúrese de que varios tiempos de ejecución sean compatibles y cambie al CRI estándar.

- Las API beta están deshabilitadas de forma predeterminada. Cuando se eliminan algunas API beta a largo plazo de Kubernetes, el 90% de los administradores de clústeres no se preocupan por las API beta. Las funciones beta no se recomiendan en el entorno de producción. Sin embargo, debido a la política de habilitación predeterminada, estas API están habilitadas en el entorno de producción, incurriendo en riesgos. Por lo tanto, en v1.24 y las versiones posteriores, las API beta están deshabilitadas de forma predeterminada, excepto para las API beta habilitadas.
- LegacyServiceAccountTokenNoAutoGeneration se mueve a la beta. De forma predeterminada, esta función está habilitada, donde no se genera automáticamente ningún token secreto para una cuenta de servicio. Si quieres usar un token que nunca caduca, necesitas crear un secreto y montarlo. Para obtener más información, consulte [Service account token Secrets](#).
- **service.alpha.kubernetes.io/tolerate-unready-endpoints** se sustituye por **Service.spec.publishNotReadyAddresses**.
- La etiqueta **Service.Spec.LoadBalancerIP** está obsoleta y puede eliminarse en versiones posteriores. Utilice una anotación personalizada.

## Referencias

Para obtener más información sobre la comparación de rendimiento y la evolución de funciones entre Kubernetes 1.25 y otras versiones, consulte los siguientes documentos:

- [Notas del lanzamiento de Kubernetes 1.25](#)
- [Notas de lanzamiento de Kubernetes 1.24](#)

### 2.1.4.2 Notas del lanzamiento de CCE Kubernetes 1.23

CCE ha aprobado el Certified Kubernetes Conformance Program y es una oferta certificada de Kubernetes. En esta sección se describen las actualizaciones de CCE Kubernetes 1.23.

## Actualizaciones de versiones

CCE proporciona optimización y actualización de componentes de enlace completo para Kubernetes 1.23.

**Tabla 2-5** Componentes principales y descripciones

Tipo de clúster	componente del núcleo	Versión	Precauciones
Clúster de CCE	Kubernetes	1.23	Ninguna
	Docker	EulerOS/CentOS: 18.9.0 Ubuntu: 18.09.9	Ninguna
	Containerd	1.4.1	Ninguna
	OS	CentOS Linux release 7.6	Ninguna
		EulerOS release 2.9	Ninguna
		EulerOS release 2.5	Ninguna
		Ubuntu 18.04 server 64bit	Ninguna
Clúster de Turbo de CCE	Kubernetes	1.23	Ninguna
	Docker	EulerOS/CentOS: 18.9.0 Ubuntu: 18.09.9	Ninguna
	Containerd	1.4.1	Ninguna
	OS (ECS)	CentOS Linux release 7.6 Ubuntu 18.04 server 64bit EulerOS release 2.9	Ninguna
		OS (BMS)	CentOS Linux release 7.6 Ubuntu 18.04 server 64bit EulerOS release 2.9

## Cambios y depreciaciones de recursos

### Cambios en CCE 1.23

- El complemento de la terminal web ya no es compatible. Use CloudShell o kubectl en su lugar.

### Notas de lanzamiento de Kubernetes 1.23

- FlexVolume está obsoleta. Use CSI.
- HorizontalPodAutoscaler v2 se promueve a GA, y la API de HorizontalPodAutoscaler v2 es estable gradualmente en la versión 1.23. No se recomienda la API HorizontalPodAutoscaler v2beta2. Usa la API v2.
- **PodSecurity** se mueve a la beta, reemplazando el PodSecurityPolicy obsoleto. PodSecurity es un controlador de admisión que aplica los estándares de seguridad de pods en los pods en el espacio de nombres basándose en etiquetas específicas del espacio de nombres que establecen el nivel de aplicación. PodSecurity está habilitado de forma predeterminada en la versión 1.23.

### Notas de lanzamiento de Kubernetes 1.22

- Las entradas ya no son compatibles con las API networking.k8s.io/v1beta1 y Extensiones/v1beta1. Si utiliza la API de una versión anterior para gestionar entradas, una aplicación no puede estar expuesta a los servicios externos. Utilice networking.k8s.io/v1.
- CustomResourceDefinitions ya no admite la API apiextensions.k8s.io/v1beta1. Si utiliza la API de una versión anterior para crear un CRD, la creación fallará, lo que afecta al controlador que reconcilia este CRD. Use apiextensions.k8s.io/v1.
- ClusterRoles, ClusterRoleBindings, Roles y RoleBindings ya no admiten la API rbac.authorization.k8s.io/v1beta1. Si utiliza la API de una versión anterior para gestionar recursos RBAC, el control de permisos de aplicación se ve afectado e incluso no puede funcionar en el clúster. Utilice rbac.authorization.k8s.io/v1.
- El ciclo de versiones de Kubernetes cambia de cuatro versiones al año a tres al año.
- StatefulSets soporte **minReadySeconds**.
- Durante el escalado, los pods se seleccionan aleatoriamente y se eliminan según el UID de pod de forma predeterminada (LogarithmicScaleDown). Esta característica mejora la aleatoriedad de los pods que se van a eliminar y alivia los problemas causados por las restricciones de propagación de la topología de pod. Para obtener más información, consulte [KEP-2185](#) e [issue 96748](#).
- La función **BoundServiceAccountTokenVolume** es estable. Esta característica mejora la seguridad del token de la cuenta de servicio y cambia el método de montaje de tokens en pods. Los clústeres de Kubernetes de v1.21 y posteriores habilitan este enfoque de forma predeterminada.

## Referencias

Para obtener más información sobre la comparación de rendimiento y la evolución de funciones entre Kubernetes 1.23 y otras versiones, consulte los siguientes documentos:

- [Notas de lanzamiento de Kubernetes v1.23](#)
- [Notas de lanzamiento de Kubernetes v1.22](#)

### 2.1.4.3 Notas del lanzamiento de CCE Kubernetes 1.21

CCE ha aprobado el Certified Kubernetes Conformance Program y es una oferta certificada de Kubernetes. En esta sección se describen las actualizaciones de CCE Kubernetes 1.21.

## Actualizaciones de versiones

CCE proporciona optimización y actualización de componentes de enlace completo para Kubernetes 1.21.

**Tabla 2-6** Componentes principales y descripciones

Tipo de clúster	componente del núcleo	Versión	Precauciones	
Clúster de CCE	Kubernetes	1.21	Ninguna	
	Docker	EulerOS/CentOS: 18.9.0 Ubuntu: 18.09.9	Ninguna	
	OS	CentOS Linux release 7.6	A partir de CCE 1.21, OverlayFS se utiliza para el almacenamiento de Docker en los nodos CentOS 7.6 en un clúster de CCE.	
		EulerOS release 2.9	Ninguna	
		EulerOS release 2.5	Ninguna	
		Ubuntu 18.04 server 64bit	Ninguna	
Clúster de Turbo de CCE	Kubernetes	1.21	Ninguna	
	Docker	EulerOS/CentOS: 18.9.0 Ubuntu: 18.09.9	Ninguna	
	Containerd	1.4.1	Ninguna	
	OS (ECS)	CentOS Linux release 7.6 Ubuntu 18.04 server 64bit EulerOS release 2.9	Ninguna	
		OS (BMS)	CentOS Linux release 7.6 Ubuntu 18.04 server 64bit EulerOS release 2.9	Ninguna

## Cambios y depreciaciones de recursos

### Notas de lanzamiento de Kubernetes 1.21

- CronJob está ahora en el estado estable, y el número de versión cambia a lote/v1.
- El Secret inmutable y ConfigMap ahora se han actualizado al estado estable. Se agrega un nuevo campo inmutable a estos objetos para rechazar los cambios. El rechazo protege a los clústeres de las actualizaciones accidentales que pueden provocar interrupciones en las aplicaciones. Como estos recursos son inmutables, kubelet no monitorea ni encuesta en busca de cambios. Esto reduce la carga de kube-apiserver y mejora la escalabilidad y el rendimiento de sus clústeres. Para obtener más información, consulte [ConfigMaps inmutable](#).
- Se ha actualizado el cierre del nodo elegante al estado de prueba. Con esta actualización, kubelet puede detectar que un nodo está apagado y terminar con gracia los pods en el nodo. Antes de esta actualización, cuando el nodo se apagó, su pod no seguía el ciclo de vida de terminación esperado, lo que causaba problemas de carga de trabajo. Ahora kubelet puede usar systemd para detectar los sistemas que están a punto de ser apagados y notificar a los pods en ejecución para que los terminen con elegancia.
- Para un pod con contenedores múltiple, puedes usar [kubectl.kubernetes.io/](#) para preseleccionar contenedores.
- PodSecurityPolicy está obsoleta. Para obtener más información, véase <https://kubernetes.io/blog/2021/04/06/podsecuritypolicy-deprecation-past-present-and-future/>.
- La función [BoundServiceAccountTokenVolume](#) ha entrado en la prueba beta. Esta característica mejora la seguridad del token de la cuenta de servicio y cambia el método de montaje de tokens en pods. Los clústeres de Kubernetes de v1.21 y posteriores habilitan este enfoque de forma predeterminada.

#### Notas de lanzamiento de Kubernetes 1.20

- La prioridad y la equidad de la API han alcanzado el estado de prueba y están habilitadas de forma predeterminada. Esto permite a kube-apiserver clasificar las solicitudes entrantes por prioridad. Para obtener más información, consulte [Prioridad y equidad de API](#).
- Se ha corregido el error de **exec probe timeouts**. Antes de que se corrija este error, la sonda exec no tiene en cuenta el campo **timeoutSeconds**. En su lugar, la sonda se ejecutará indefinidamente, incluso más allá de su fecha límite configurada. Se detendrá hasta que se devuelva el resultado. Ahora, si no se especifica ningún valor, se utiliza el valor predeterminado, es decir, un segundo. Si el tiempo de detección supera un segundo, la comprobación de estado de la aplicación puede fallar. Actualice el campo **timeoutSeconds** para las aplicaciones que utilizan esta función durante la actualización. La reparación proporcionada por la nueva función de gating de ExecProbeTimeout permite al operador del clúster restaurar el comportamiento anterior, pero este comportamiento se bloqueará y eliminará en versiones posteriores.
- RuntimeClass entra en el estado estable. RuntimeClass proporciona un mecanismo para admitir varios tiempos de ejecución en un clúster y exponer información sobre el tiempo de ejecución contenedor al plano de control.
- La depuración de kubectl ha alcanzado el estado de prueba. La depuración de kubectl proporciona soporte para flujos de trabajo de depuración comunes.
- Dockershim fue marcado como obsoleto en Kubernetes 1.20. Actualmente, puede seguir usando Docker en el clúster. Este cambio es irrelevante para la imagen de contenedor utilizada por los clústeres. Aún puede usar Docker para crear sus imágenes. Para obtener más información, consulte [Preguntas frecuentes sobre la deprecación de Dockershim](#).

## Referencias

Para obtener más información sobre la comparación de rendimiento y la evolución de funciones entre Kubernetes 1.21 y otras versiones, consulte los siguientes documentos:

- [Notas de lanzamiento de Kubernetes v1.21](#)
- [Notas de lanzamiento de Kubernetes v1.20](#)

### 2.1.4.4 Kubernetes 1.19 (EOM) Release Notes

CCE ha aprobado el Certified Kubernetes Conformance Program y es una oferta certificada de Kubernetes. En esta sección se describen las actualizaciones de CCE Kubernetes 1.19.

## Cambios y depreciaciones de recursos

### Notas de la versión de Kubernetes 1.19

- Los volúmenes en árbol de vSphere se pueden migrar a los controladores CSI de vSphere. El complemento in-tree vSphere Volume ya no se usa y se eliminará en las versiones posteriores.
- [apiextensions.k8s.io/v1beta1](#) ha sido obsoleta. Use [apiextensions.k8s.io/v1](#) en su lugar.
- [apiregistration.k8s.io/v1beta1](#) ha sido obsoleta. Use [apiregistration.k8s.io/v1](#) en su lugar.
- [authentication.k8s.io/v1beta1](#) y [authorization.k8s.io/v1beta1](#) han sido obsoletas y se quitarán de Kubernetes 1.22. Use [authentication.k8s.io/v1](#) y [authorization.k8s.io/v1](#) en su lugar.
- [autoscaling/v2beta1](#) ha sido obsoleta. Use [autoscaling/v2beta2](#) en su lugar.
- [coordination.k8s.io/v1beta1](#) ha sido obsoleta en Kubernetes 1.19 y se quitará de la versión 1.22. Use [coordination.k8s.io/v1](#) en su lugar.
- kube-apiserver: La API **componentstatus** ha sido obsoleta.
- kubeadm: El comando **kubeadm config view** ha sido obsoleta y se eliminará en las versiones posteriores. Utilice **kubectrl get cm -o yaml -n kube-system kubeadm-config** para obtener directamente la configuración kubeadm.
- kubeadm: El comando **kubeadm alpha kubelet config enable-dynamic** ha sido obsoleta.
- kubeadm: El indicador **--use-api** en el comando **kubeadm alpha certs renew** ha sido obsoleta.
- Kubernetes ya no admite la creación de imágenes **hyperkube**.
- El indicador **--export** se quita del comando **kubectrl get**.
- Se ha eliminado el **ResourceLimitsPriorityFunction** de función alfa.
- [storage.k8s.io/v1beta1](#) ha sido obsoleta. Use [storage.k8s.io/v1](#) en su lugar.

### Notas de la versión de Kubernetes 1.18

- kube-apiserver
  - Todos los recursos de las versiones de API [apps/v1beta1](#) y [apps/v1beta2](#) ya no se sirven. Puede usar la versión de la API [apps/v1](#).
  - DaemonSets, Deployments y ReplicaSets en la versión de la API [extensions/v1beta1](#) ya no se sirven. Puede usar la versión de la API [apps/v1](#).

- Las NetworkPolicies en la versión de la API **extensions/v1beta1** ya no se sirven. Puede usar la versión de la API **networking.k8s.io/v1**.
  - Las PodSecurityPolicies en la versión de la API **extensions/v1beta1** ya no se sirven. Migre para usar la versión de la API **policy/v1beta1**.
  - kubelet
    - **--redirect-container-streaming** no se recomienda y será obsoleta en v1.20.
    - El punto de medición de recursos **/metrics/resource/v1alpha1** y todas las normas de medición bajo este punto de conexión han sido obsoletas. Utilice los estándares de medición bajo el punto de conexión **/metrics/resource** en su lugar:
      - `scrape_error --> scrape_error`
      - `node_cpu_usage_seconds_total --> node_cpu_usage_seconds`
      - `node_memory_working_set_bytes --> node_memory_working_set_bytes`
      - `container_cpu_usage_seconds_total --> container_cpu_usage_seconds`
      - `container_memory_working_set_bytes --> container_memory_working_set_bytes`
      - `scrape_error --> scrape_error`
    - En futuras versiones, kubelet ya no creará el directorio de destino **CSI NodePublishVolume** de acuerdo con las especificaciones de CSI. Es posible que necesite actualizar el controlador de CSI en consecuencia para crear y procesar correctamente la ruta de destino.
  - kube-proxy
    - No se recomienda utilizar las banderas **--healthz-port** y **--metrics-port**. Use **--healthz-bind-address** y **--metrics-bind-address** en su lugar.
    - La opción de función **EndpointSliceProxying** se agrega para controlar el uso de EndpointSlices en kube-proxy. Esta función está deshabilitada por defecto.
  - kubeadm
    - La marca **--kubelet-version** de **kubeadm upgrade node** ha sido obsoleta y se eliminará en versiones posteriores.
    - La marca **--use-api** del comando **kubeadm alpha certs renew** ha sido obsoleta.
    - kube-dns ha sido obsoleta y ya no será compatible en versiones futuras.
    - La estructura de ClusterStatus en el ConfigMap **kubeadm-config** ha sido obsoleta y se eliminará en versiones posteriores.
  - kubectl
    - No se recomienda utilizar valores boolean y unset para **--dry-run**. **server|client|none** se utiliza en la nueva versión.
    - **--server-dry-run** ha sido obsoleta para **kubectl apply** y reemplazado por **--dry-run=server**.
  - Complementos
- Se elimina la supervisión del clúster.
- kube-scheduler
    - La métrica **scheduling\_duration\_seconds** ha sido obsoleta.
    - Las métricas **scheduling\_algorithm\_predicate\_evaluation\_seconds** y **scheduling\_algorithm\_priority\_evaluation\_seconds counters** ya no se usan y son reemplazadas por

- `framework_extension_point_duration_seconds[extension_point="Filter"]` y `framework_extension_point_duration_seconds[extension_point="Score"]`.
- La política del programador AlwaysCheckAllPredictes ha sido obsoleta.
- Otros cambios
  - El componente `k8s.io/node-api` ya no se actualiza. En su lugar, puede utilizar el tipo `RuntimeClass` de `k8s.io/api` y los clientes generados de `k8s.io/client-go`.
  - La etiqueta `client` se ha eliminado de `apiserver_request_total`.

## Referencias

Para obtener más información sobre la comparación de rendimiento y la evolución de funciones entre Kubernetes 1.19 y otras versiones, consulte los siguientes documentos:

- [Notas de la versión de Kubernetes v1.19.0](#)
- [Notas de la versión de Kubernetes v1.18.0](#)

### 2.1.4.5 Kubernetes 1.17 (EOM) Release Notes

CCE ha aprobado el Certified Kubernetes Conformance Program y es una oferta certificada de Kubernetes. En esta sección se describen las actualizaciones de CCE Kubernetes 1.17.

## Cambios y depreciaciones de recursos

- Todos los recursos de las versiones de API `apps/v1beta1` y `apps/v1beta2` ya no se sirven. Migre para usar la versión de la API `apps/v1`.
- `DaemonSets`, `Deployments` y `ReplicaSets` en la versión de la API `extensions/v1beta1` ya no se sirven. Puede usar la versión de la API `apps/v1`.
- Las `NetworkPolicies` en la versión de la API `extensions/v1beta1` ya no se sirven. Migre para usar la versión de la API `networking.k8s.io/v1`.
- Las `PodSecurityPolicies` en la versión de la API `extensions/v1beta1` ya no se sirven. Migre para usar la versión de la API `policy/v1beta1`.
- `Ingress` en la versión de la API `extensions/v1beta1` ya no se servirán en v1.20. Migre para usar la versión de la API `networking.k8s.io/v1beta1`.
- `PriorityClass` en las versiones de API `scheduling.k8s.io/v1beta1` y `scheduling.k8s.io/v1alpha1` ya no se sirven en v1.17. Migre para usar la versión de la API `scheduling.k8s.io/v1`.
- El campo `event series.state` de la versión de la API `events.k8s.io/v1beta1` ha sido obsoleta y se quitará de v1.18.
- `CustomResourceDefinition` en la versión de la API `apiextensions.k8s.io/v1beta1` ha sido obsoleta y ya no se servirá en v1.19. Use la versión de la API `apiextensions.k8s.io/v1`.
- `MutatingWebhookConfiguration` y `ValidatingWebhookConfiguration` en la versión de la API `admissionregistration.k8s.io/v1beta1` han sido obsoletas y ya no se servirán en la v1.19. Puede usar la versión de la API `admissionregistration.k8s.io/v1`.
- Las versiones de API `rbac.authorization.k8s.io/v1alpha1` y `rbac.authorization.k8s.io/v1beta1` han sido obsoletas y ya no se servirán en v1.20. Use la versión de la API `rbac.authorization.k8s.io/v1`.
- El objeto `CSINode` de `storage.k8s.io/v1beta1` ha sido obsoleta y se quitará en versiones posteriores.

## Otras depreciaciones y extracciones

- **OutOfDisk node condition** se quita a favor de **DiskPressure**.
- La anotación **scheduler.alpha.kubernetes.io/critical-pod** se quita a favor de **priorityClassName**.
- **beta.kubernetes.io/os** y **beta.kubernetes.io/arch** han sido obsoletas en v1.14 y se quitarán en v1.18.
- No utilice **--node-labels** para establecer etiquetas con el prefijo **kubernetes.io** y **k8s.io**. La etiqueta **kubernetes.io/availablezone** en versiones anteriores se quita en v1.17 y se cambia a **failure-domain.beta.kubernetes.io/zone**.
- El **beta.kubernetes.io/instance-type** está obsoleta a favor de **node.kubernetes.io/instance-type**.
- Quite la ruta **{kubelet\_root\_dir}/plugins**.
- Quite las funciones de clúster integradas **system:csi-external-provisioner** y **system:csi-external-attacher**.

## Referencias

Para obtener más información sobre la comparación de rendimiento y la evolución de funciones entre Kubernetes 1.17 y otras versiones, consulte los siguientes documentos:

- [Notas de la versión de Kubernetes v1.17.0](#)
- [Notas de la versión de Kubernetes v1.16.0](#)

### 2.1.4.6 Kubernetes 1.15 (EOM) Release Notes

CCE ha aprobado el Certified Kubernetes Conformance Program y es una oferta certificada de Kubernetes. En esta sección se describen las actualizaciones de CCE Kubernetes 1.15.

Actualice sus clústeres de Kubernetes antes de la versión EOM para una ejecución más estable y confiable del clúster.

## Descripción

CCE proporciona optimización y actualización de componentes de enlace completo para Kubernetes v1.15, que incluye dos versiones secundarias v1.15.11 y v1.15.6-r1.

## Cambios y depreciaciones de recursos

- La entrada en la versión de la API **extensions/v1beta1** ha sido obsoleta. Ya no se servirá desde Kubernetes 1.19. Puede usar la versión de la API **networking.k8s.io/v1beta1**.
- NetworkPolicy en la versión de API **extensions/v1beta1** será oficialmente suspendido en 1.16. Migre para usar la versión de la API **networking.k8s.io/v1**.
- PodSecurityPolicy en la versión de API **extensions/v1beta1** será oficialmente suspendido en 1.16. Migre para usar la versión de la API **policy/v1beta1**.
- DaemonSets, Deployments y ReplicaSets en las versiones de la API **extensions/v1beta1**, **apps/v1beta1** y **apps/v1beta2** no se servirán en 1.16. Puede usar la versión de la API **apps/v1**.
- PriorityClass se actualiza a **scheduling.k8s.io/v1**, **scheduling.k8s.io/v1beta1** y **scheduling.k8s.io/v1alpha1**. Será obsoleta en 1.17.

- El campo **series.state** de la versión de la Event API **events.k8s.io/v1beta1** ha sido obsoleta y se quitará de 1.18.

## Referencias

Log de cambios de v1.13 a v1.15

- Log de cambios de v1.14 a v1.15:  
<https://github.com/kubernetes/kubernetes/blob/master/CHANGELOG/CHANGELOG-1.15.md>
- Log de cambios de v1.13 a v1.14:  
<https://github.com/kubernetes/kubernetes/blob/master/CHANGELOG/CHANGELOG-1.14.md>

### 2.1.4.7 Kubernetes 1.13 (EOM) Release Notes

CCE ha aprobado el Certified Kubernetes Conformance Program y es una oferta certificada de Kubernetes. En esta sección se describen las actualizaciones de CCE Kubernetes 1.13.

**Tabla 2-7** Descripción de la versión 1.13

Kubernetes (versión mejorada de CCE)	Descripción
v1.13.10-r0	<p><b>Información destacada:</b></p> <ul style="list-style-type: none"> <li>● Los nodos de Arm se pueden agregar a un clúster de CCE.</li> <li>● El nombre del balanceador de carga es configurable.</li> <li>● El equilibrio de carga de capa 4 admite la comprobación de estado y el equilibrio de carga de capa 7 admite la comprobación de estado, la política de asignación y la sesión adhesiva.</li> <li>● Los nodos de BMS se pueden crear en un clúster de CCE (cuando se utiliza el modelo de red de túnel).</li> <li>● Los nodos acelerados por Ascend (soportados por los procesadores de AI de HiSilicon Ascend 310) se aplican a escenarios como el reconocimiento de imágenes, el procesamiento de vídeo, el cómputo de inferencia y el aprendizaje automático.</li> <li>● El baseSize de docker es configurable.</li> <li>● Se admite la programación de afinidad de espacio de nombres.</li> <li>● El espacio de usuario se puede particionar en discos de datos de nodo.</li> <li>● Se pueden configurar las políticas de gestión de CPU de clúster.</li> <li>● Los nodos de un clúster se pueden configurar por las subredes (cuando se utiliza el modo de red de túnel).</li> </ul>

Kubernetes (versión mejorada de CCE)	Descripción
v1.13.7-r0	<b>Información destacada:</b> <ul style="list-style-type: none"> <li>● Se incorporan las características de Kubernetes v1.13.7.</li> <li>● Se admite la definición de conexión de red.</li> </ul>

## Referencias

Registro de cambios de v1.11 a v1.13

- Registro de cambios de v1.12 a v1.13:  
<https://github.com/kubernetes/kubernetes/blob/master/CHANGELOG/CHANGELOG-1.13.md>
- Registro de cambios de v1.11 a v1.12:  
<https://github.com/kubernetes/kubernetes/blob/master/CHANGELOG/CHANGELOG-1.12.md>

### 2.1.4.8 Kubernetes 1.11 (EOM) Release Notes

CCE ha aprobado el Certified Kubernetes Conformance Program y es una oferta certificada de Kubernetes. En esta sección se describen las actualizaciones de CCE Kubernetes 1.11.

**Tabla 2-8** Descripción de la versión 1.11

Kubernetes (versión mejorada de CCE)	Descripción
v1.11.7-r2	<b>Información destacada:</b> <ul style="list-style-type: none"> <li>● Se proporciona soporte para GPU V100.</li> <li>● Se proporciona soporte para la gestión de permisos.</li> </ul>

Kubernetes (versión mejorada de CCE)	Descripción
v1.11.7-r0	<p><b>Información destacada:</b></p> <ul style="list-style-type: none"> <li>● Se incorporan las características de Kubernetes v1.11.7.</li> <li>● Se pueden crear grupos de nodos, máquinas virtuales y clústeres de Kungpeng.</li> <li>● Los nodos de BMS se pueden crear en un clúster de CCE (cuando se utiliza el modelo de red de VPC), y se admite el despliegue híbrido de BMS y VM.</li> <li>● Se proporciona soporte para GPU V100.</li> <li>● Application Operations Management (AOM) notifica a los usuarios cuando se generan alarmas para clústeres de contenedor de v1.11.</li> <li>● Se admite la conmutación de tipo de acceso para los Services.</li> <li>● Se pueden configurar segmentos de red de Service.</li> <li>● Se puede personalizar el número de direcciones IP asignadas a un nodo de un clúster.</li> </ul>
v1.11.3-r2	<p><b>Información destacada:</b></p> <ul style="list-style-type: none"> <li>● Los clústeres admiten IPv6 de doble pila.</li> <li>● Algoritmos de equilibrio de carga de ELB: hash IP de origen y sesiones adhesivas con servidores backend.</li> </ul>
v1.11.3-r1	<p><b>Información destacada:</b></p> <ul style="list-style-type: none"> <li>● Las expresiones regulares de Perl se pueden usar para coincidir con los URL de ingress.</li> </ul>
v1.11.3-r0	<p><b>Información destacada:</b></p> <ul style="list-style-type: none"> <li>● Se incorporan las características de Kubernetes v1.11.3.</li> <li>● Los nodos maestros de un clúster se pueden desplegar a través de múltiples AZ.</li> <li>● CCE trabaja con SFS Turbo para proporcionar almacenamiento contenedor.</li> </ul>

## Referencias

Log de cambios de v1.9 a v1.11

- Log de cambios de v1.10 a v1.11:  
<https://github.com/kubernetes/kubernetes/blob/master/CHANGELOG/CHANGELOG-1.11.md>
- Log de cambios de v1.9 a v1.10:  
<https://github.com/kubernetes/kubernetes/blob/master/CHANGELOG/CHANGELOG-1.10.md>

## 2.1.4.9 Release Notes for Kubernetes 1.9 (EOM) and Earlier Versions

CCE ha aprobado el Certified Kubernetes Conformance Program y es una oferta certificada de Kubernetes. En esta sección se describen las actualizaciones de CCE Kubernetes 1.9 y versiones anteriores.

**Tabla 2-9** Descripción de v1.9 y versiones anteriores

Kubernetes (versión mejorada de CCE)	Descripción
v1.9.10-r2	<p><b>Información destacada:</b></p> <ul style="list-style-type: none"> <li>● Algoritmos de equilibrio de carga de ELB: hash IP de origen y sesiones adhesivas con servidores backend.</li> </ul>
v1.9.10-r1	<p><b>Información destacada:</b></p> <ul style="list-style-type: none"> <li>● CCE trabaja con SFS.</li> <li>● Los ELB mejorados se pueden crear automáticamente para los Services.</li> <li>● Se admite la transmisión transparente de direcciones IP de origen para los ELB mejorados en la red pública.</li> <li>● Se puede configurar el número máximo de pods en un nodo.</li> </ul>
v1.9.10-r0	<p><b>Información destacada:</b></p> <ul style="list-style-type: none"> <li>● Uso de ELB/ingress para clústeres de Kubernetes; nuevo mecanismo de control de tráfico</li> <li>● Se incorporan las características de Kubernetes v1.9.10.</li> <li>● Se admite la autorización de capacidad de Kubernetes RBAC.</li> </ul> <p><b>Rectificación de fallos:</b></p> <ul style="list-style-type: none"> <li>● Pérdida de memoria ocasional en los nodos, que es causada por errores del kernel cgroup</li> </ul>
v1.9.7-r1	<p><b>Información destacada:</b></p> <ul style="list-style-type: none"> <li>● Se mejora el mecanismo de notificación de eventos de PVC y de PersistentVolume (PV). Los eventos se pueden ver en la página de detalles de PVC.</li> <li>● CCE funciona con un sistema de autenticación de terceros.</li> <li>● Las máquinas físicas que usan EulerOS 2.3 pueden ser gestionadas.</li> <li>● La asignación del disco de datos puede definirse por el usuario.</li> <li>● Los discos de Elastic Volume Service (EVS) son compatibles con los BMS.</li> <li>● Las NIC InfiniBand son compatibles con los BMS.</li> <li>● Los nodos se pueden crear usando la API CM-v3 en escenarios de BMS.</li> </ul>

Kubernetes (versión mejorada de CCE)	Descripción
v1.9.7-r0	<p><b>Información destacada:</b></p> <ul style="list-style-type: none"> <li>● La versión de Docker de los nuevos clústeres se ha actualizado a v17.06.</li> <li>● Se admite la conexión en cascada de DNS.</li> <li>● Se pueden gestionar complementos.</li> <li>● Se incorporan las características de Kubernetes v1.9.7.</li> <li>● Se admite el HTTPS de ingress de capa-7.</li> <li>● StatefulSets se puede migrar, programar, actualizar y actualizar.</li> </ul>
v1.9.2-r3	<p><b>Información destacada:</b></p> <ul style="list-style-type: none"> <li>● Los nodos de clúster que usan CentOS 7.4 pueden crearse o gestionarse.</li> <li>● Los servicios de DNAT son compatibles.</li> <li>● Se proporcionan las API de NetworkPolicy.</li> <li>● Se pueden configurar varios puertos para un Service de Kubernetes que utiliza un ELB.</li> </ul> <p><b>Rectificación de fallos:</b></p> <ul style="list-style-type: none"> <li>● Reciclaje incompleto de recursos de pod causado por una desconexión con kube-apiserver</li> <li>● Inexactitud de los datos durante el ajuste automático de nodos</li> </ul>
v1.9.2-r2	<p><b>Información destacada:</b></p> <ul style="list-style-type: none"> <li>● Los puertos de comprobación de estado personalizados se pueden configurar para los balanceadores de carga clásicos.</li> <li>● Se mejora el rendimiento de los balanceadores de carga clásicos.</li> <li>● Los puertos de Kubernetes Service se pueden configurar para el equilibrio de carga de capa 4.</li> </ul> <p><b>Rectificación de fallos:</b></p> <ul style="list-style-type: none"> <li>● Errores en los complementos de red, que causan interbloqueos en las comprobaciones de estado</li> <li>● Un número limitado de conexiones de HAProxy en un clúster HA</li> </ul>
v1.9.2-r1	<p><b>Información destacada:</b></p> <ul style="list-style-type: none"> <li>● Se incorporan las características de Kubernetes v1.9.2.</li> <li>● Los nodos de clúster admiten CentOS 7.1.</li> <li>● Los nodos de GPU son compatibles y el uso de recursos de GPU puede restringirse.</li> <li>● Se admite el complemento de la terminal web.</li> </ul>

Kubernetes (versión mejorada de CCE)	Descripción
v1.7.3-r13	<p><b>Información destacada:</b></p> <ul style="list-style-type: none"> <li>● La versión de Docker de los nuevos clústeres se ha actualizado a v17.06.</li> <li>● Se admite la conexión en cascada de DNS.</li> <li>● Se pueden gestionar complementos.</li> <li>● Se mejora el mecanismo de notificación de eventos de PVC y PV.</li> <li>● OBS es compatible con clústeres de BMS.</li> </ul>
v1.7.3-r12	<p><b>Información destacada:</b></p> <ul style="list-style-type: none"> <li>● Los nodos de clúster que usan CentOS 7.4 pueden crearse o gestionarse.</li> <li>● Los servicios de DNAT son compatibles.</li> <li>● Se proporcionan las API de NetworkPolicy.</li> <li>● Se pueden configurar varios puertos para un Service de Kubernetes que utiliza un ELB.</li> </ul> <p><b>Rectificación de fallos:</b></p> <ul style="list-style-type: none"> <li>● Reciclaje incompleto de recursos de pod causado por una desconexión con kube-apiserver</li> <li>● Inexactitud de los datos durante el ajuste automático de nodos</li> <li>● Se modifica la solicitud de período de caducidad del evento. El período de envejecimiento del grupo es de 1 hora.</li> </ul>
v1.7.3-r11	<p><b>Información destacada:</b></p> <ul style="list-style-type: none"> <li>● Los puertos de comprobación de estado personalizados se pueden configurar para los balanceadores de carga clásicos.</li> <li>● Se mejora el rendimiento de los balanceadores de carga clásicos.</li> <li>● Los puertos de Kubernetes Service se pueden configurar para el equilibrio de carga de capa 4.</li> <li>● Los espacios de nombres se pueden eliminar.</li> <li>● Los disco de EVS pueden desvincularse.</li> <li>● Se pueden configurar las políticas de migración.</li> </ul> <p><b>Rectificación de fallos:</b></p> <ul style="list-style-type: none"> <li>● Errores en los complementos de red, que causan interbloqueos en las comprobaciones de estado</li> <li>● Un número limitado de conexiones de HAProxy en un clúster HA</li> </ul>

Kubernetes (versión mejorada de CCE)	Descripción
v1.7.3-r10	<p><b>Información destacada:</b></p> <ul style="list-style-type: none"> <li>● Se admiten redes de contenedor L2 superpuestas.</li> <li>● Los nodos de clúster pueden ser máquinas virtuales aceleradas por GPU.</li> <li>● Los nodos de clúster admiten CentOS 7.1 y se puede seleccionar el sistema operativo.</li> <li>● Los clústeres de Windows admiten ELB.</li> <li>● Los nodos de CCE pueden usar SFS para almacenamiento.</li> <li>● Los clústeres de BMS admiten SFS.</li> </ul>
v1.7.3-r9	<p><b>Información destacada:</b></p> <ul style="list-style-type: none"> <li>● El despliegue entre las AZ es compatible con cargas de trabajo.</li> <li>● Los contenedores soportan OBS.</li> <li>● Se admite el equilibrio de carga de capa 7.</li> <li>● Los clústeres de Windows admiten EVS.</li> <li>● El mapeador de dispositivos en modo direct-lvm es compatible con escenarios de BMS.</li> </ul>
v1.7.3-r8	<p><b>Información destacada:</b></p> <ul style="list-style-type: none"> <li>● Se admite el ajuste automático para los nodos de clúster.</li> <li>● Los nodos del Arm pueden ser gestionados.</li> </ul>
v1.7.3-r7	<p><b>Información destacada:</b></p> <ul style="list-style-type: none"> <li>● Los nodos sp2 de SUSE 12 se pueden gestionar en los clústeres de contenedor (en el modo de red de túnel).</li> <li>● Docker soporta el mapeador de dispositivos en modo direct-lvm.</li> <li>● Los clústeres admiten el complemento del panel de control.</li> <li>● Se pueden crear clústeres de Windows.</li> </ul>
v1.7.3-r6	<p><b>Información destacada:</b></p> <ul style="list-style-type: none"> <li>● Las API de EVS nativas son compatibles con los clústeres.</li> </ul>
v1.7.3-r5	<p><b>Información destacada:</b></p> <ul style="list-style-type: none"> <li>● Se pueden crear clústeres de HA.</li> </ul> <p><b>Rectificación de fallos:</b></p> <ul style="list-style-type: none"> <li>● Desconexión de la red del contenedor después del reinicio de un nodo</li> </ul>
v1.7.3-r4	<p><b>Información destacada:</b></p> <ul style="list-style-type: none"> <li>● Se mejora el rendimiento del clúster.</li> <li>● La interconexión con ELB está permitida en escenarios de BMS.</li> </ul>

Kubernetes (versión mejorada de CCE)	Descripción
v1.7.3-r3	<b>Información destacada:</b> <ul style="list-style-type: none"> <li>● El almacenamiento se puede conectar a máquinas virtuales basadas en el núcleo (KVM).</li> </ul>
v1.7.3-r2	<b>Información destacada:</b> <ul style="list-style-type: none"> <li>● Se admite SFS para proporcionar almacenamiento de contenedor.</li> <li>● Los registros personalizados se pueden configurar para cargas de trabajo.</li> <li>● Se admite la reducción agraciada para las cargas de trabajo.</li> </ul> <b>Rectificación de fallos:</b> <ul style="list-style-type: none"> <li>● Expiración de ID de clave de acceso/clave de acceso secreto (AK/SK) de volúmenes de almacenamiento de contenedor</li> </ul>
v1.7.3-r1	<b>Información destacada:</b> <ul style="list-style-type: none"> <li>● Los nombres de dominio externos pueden ser resueltos por kube-dns.</li> </ul>
v1.7.3-r0	<b>Información destacada:</b> <ul style="list-style-type: none"> <li>● Se incorporan las características de Kubernetes v1.7.3.</li> <li>● Se admite Elastic Load Balance (ELB).</li> <li>● El almacenamiento se puede conectar a las máquinas virtuales de Xen.</li> <li>● Se admite EVS para proporcionar almacenamiento de contenedor.</li> </ul>

## 2.1.5 Notas de la versión de clúster

Cada versión de clúster se adapta a múltiples sistemas operativos. Para obtener más información, véase [Descripción del nodo del SO](#).

## Versión 1.25

**Tabla 2-10** Notas de la versión del parche v1.25

Versión del parche del clúster de CCE	Versión de Kubernetes	Actualizaciones de funciones	Optimización	Fijación de vulnerabilidades
v1.25.3-r0	<a href="#">v1.25.5</a>	<ul style="list-style-type: none"> <li>● Los servicios y las entradas soportan los balanceadores de carga dedicados que han estado vinculados a los GEIP.</li> <li>● Los ENIs de CCE Turbo admiten direcciones IP fijas.</li> <li>● CCE Turbo ENIs soporta la creación y encuadernación automática de EIPs.</li> <li>● Mejora el despliegue híbrido de los clústeres de Turbo de CCE: Se admite la restricción de prioridad de red de pod.</li> <li>● Los clústeres de Turbo de CCE admiten la asociación entre espacios de nombres y bloques CIDR de contenedor.</li> <li>● Admite CPU Burst para evitar que la limitación del tráfico de la CPU afecte a los servicios sensibles a la latencia.</li> </ul>	Mejora la estabilidad de la red cuando se cambian las especificaciones de los clústeres de CCE Turbo.	Se han corregido algunos problemas de seguridad.

Versión del parche del clúster de CCE	Versión de Kubernetes	Actualizaciones de funciones	Optimización	Fijación de vulnerabilidades
v1.25.1-r0	<a href="#">v1.25.5</a>	El clúster CCE v1.25 se publica por primera vez. Para obtener más información, consulte <a href="#">Notas del lanzamiento de CCE Kubernetes 1.25</a> .	-	-

## Versión 1.23

**Tabla 2-11** Notas de la versión del parche v1.23

Versión del parche del clúster de CCE	Versión de Kubernetes	Actualizaciones de funciones	Optimización	Fijación de vulnerabilidades
v1.23.8-r0	<a href="#">v1.23.11</a>	<ul style="list-style-type: none"> <li>● Los servicios y las entradas soportan los balanceadores de carga dedicados que han estado vinculados a los GEIP.</li> <li>● Los ENIs de CCE Turbo admiten direcciones IP fijas.</li> <li>● CCE Turbo ENIs soporta la creación y encuadernación automática de EIPs.</li> <li>● Mejora el despliegue híbrido de los clústeres de Turbo de CCE: Se admite la restricción de prioridad de red de pod.</li> <li>● Los clústeres de Turbo de CCE admiten la asociación entre espacios de nombres y bloques CIDR de contenedor.</li> <li>● Admite CPU Burst para evitar que la limitación del tráfico de la CPU afecte a los servicios sensibles a la latencia.</li> </ul>	<ul style="list-style-type: none"> <li>● Mejora la confiabilidad de Docker durante la actualización.</li> <li>● Optimiza la sincronización de tiempo de los nodos.</li> </ul>	Se han corregido algunos problemas de seguridad.

Versión del parche del clúster de CCE	Versión de Kubernetes	Actualizaciones de funciones	Optimización	Fijación de vulnerabilidades
v1.23.7-r10	<a href="#">v1.23.11</a>	-	<ul style="list-style-type: none"> <li>● Mejora la confiabilidad de Docker durante la actualización.</li> <li>● Mejora la fiabilidad de containerd en caso de desconexiones.</li> <li>● Soporta endurecimiento de seguridad en escenarios cuando se optimizan los parámetros del núcleo.</li> </ul>	Se han corregido algunos problemas de seguridad.
v1.23.7-r0	<a href="#">v1.23.11</a>	<ul style="list-style-type: none"> <li>● Los servicios y las entradas soportan los balanceadores de carga dedicados que han estado vinculados a los GEIP.</li> </ul>	<ul style="list-style-type: none"> <li>● Mejora la estabilidad de la red cuando se cambian las especificaciones de los clústeres de CCE Turbo.</li> <li>● Mejora la estabilidad de la red de nginx-ingress-controller en escenarios de actualización de clúster.</li> <li>● Optimiza la sincronización de tiempo de los nodos.</li> </ul>	Se han corregido algunos problemas de seguridad.

Versión del parche del clúster de CCE	Versión de Kubernetes	Actualizaciones de funciones	Optimización	Fijación de vulnerabilidades
v1.23.6-r0	<a href="#">v1.23.11</a>	<ul style="list-style-type: none"> <li>● Los puertos TCP/UDP se pueden configurar para los servicios de LoadBalancer al mismo tiempo. Para obtener más información, véase <a href="#">Configuración de la comprobación de estado para varios puertos</a>.</li> <li>● Puerta de preparación de pods soportada. Para obtener más información, véase <a href="#">Configuración del estado del pod a través de la comprobación de estado de ELB</a>.</li> </ul>	<ul style="list-style-type: none"> <li>● Se ha mejorado la fiabilidad de la tabla de flujo cuando la red subyacente es anormal.</li> <li>● Se ha mejorado la estabilidad del sistema operativo con una versión posterior del núcleo en escenarios de reinicio, como un apagado inesperado.</li> <li>● Métricas de GPU/NPU cadvisor optimizadas.</li> </ul>	Se han corregido algunos problemas de seguridad.

Versión del parche del clúster de CCE	Versión de Kubernetes	Actualizaciones de funciones	Optimización	Fijación de vulnerabilidades
v1.23.5-r0	<a href="#">v1.23.11</a>	<ul style="list-style-type: none"> <li>● El almacenamiento de contenedores puede interconectarse con SFS 3.0.</li> <li>● Soporta la detección de fallas del dispositivo y el aislamiento de los nodos de la GPU.</li> <li>● Admite grupos de seguridad personalizados por clúster.</li> <li>● Los ENIs Trunkport a nivel de nodo pueden estar preenlazados para los clústeres de CCE Turbo.</li> <li>● Puede recopilar logs de plano de control.</li> <li>● Huawei Cloud EulerOS 2.0 desarrollado por Huawei es compatible.</li> <li>● containerd es compatible.</li> <li>● Los clústeres de CCE Turbo admiten la afinidad híbrida del despliegue y de la CPU.</li> </ul>	<ul style="list-style-type: none"> <li>● La versión de ETCD del nodo principal se actualiza a la versión 3.5.6 de Kubernetes.</li> <li>● El rendimiento de acceso al Service en los nodos de EulerOS 2.8 está optimizado.</li> <li>● La programación está optimizada. Los pods se distribuyen uniformemente a través de AZ cuando los pods se escalan.</li> <li>● El uso de memoria de kube-apiserver se optimiza cuando los CRD se actualizan con frecuencia.</li> </ul>	<p>Se han corregido algunos problemas de seguridad y las siguientes vulnerabilidades de CVE:</p> <ul style="list-style-type: none"> <li>● <a href="#">CVE-2022-3294</a></li> <li>● <a href="#">CVE-2022-3162</a></li> <li>● <a href="#">CVE-2022-3172</a></li> <li>● <a href="#">CVE-2021-25749</a></li> </ul>
v1.23.4-r10	<a href="#">v1.23.4</a>	-	El uso de memoria de kube-apiserver se optimiza cuando los CRD se actualizan con frecuencia.	Se han corregido algunos problemas de seguridad.

Versión del parche del clúster de CCE	Versión de Kubernetes	Actualizaciones de funciones	Optimización	Fijación de vulnerabilidades
v1.23.4-r0	<a href="#">v1.23.4</a>	Soporta la creación de nodos de Arm.	-	Se han corregido algunos problemas de seguridad.
v1.23.1-r0	<a href="#">v1.23.4</a>	El clúster de CCE v1.23 se publica por primera vez. Para obtener más información, consulte <a href="#">Notas del lanzamiento de CCE Kubernetes 1.23</a> .	-	-

## Version 1.21

Tabla 2-12 Notas de la versión del parche v1.21

Versión del parche del clúster de CCE	Versión de Kubernetes	Actualizaciones de funciones	Optimización	Fijación de vulnerabilidades
v1.21.10-r0	<a href="#">v1.21.14</a>	<ul style="list-style-type: none"> <li>Los servicios y las entradas soportan los balanceadores de carga dedicados que han estado vinculados a los GEIP.</li> <li>Los ENIs de CCE Turbo admiten direcciones IP fijas.</li> <li>CCE Turbo ENIs soporta la creación y encuadernación automática de EIPs.</li> </ul>	<ul style="list-style-type: none"> <li>Mejora la confiabilidad de Docker durante la actualización.</li> <li>Optimiza la sincronización de tiempo de los nodos.</li> <li>Optimiza la estabilidad de la extracción de imágenes cuando Docker se está ejecutando después de reiniciar el nodo.</li> </ul>	Se han corregido algunos problemas de seguridad.

Versión del parche del clúster de CCE	Versión de Kubernetes	Actualizaciones de funciones	Optimización	Fijación de vulnerabilidades
v1.21.9-r0	<a href="#">v1.21.14</a>	<ul style="list-style-type: none"> <li>● Los servicios y las entradas soportan los balanceadores de carga dedicados que han estado vinculados a los GEIP.</li> </ul>	Mejora la estabilidad de la red cuando se cambian las especificaciones de los clústeres de CCE Turbo.	Se han corregido algunos problemas de seguridad.
v1.21.8-r0	<a href="#">v1.21.14</a>	<ul style="list-style-type: none"> <li>● Los puertos TCP/UDP se pueden configurar para los servicios de LoadBalancer al mismo tiempo. Para obtener más información, véase <a href="#">Configuración de la comprobación de estado para varios puertos</a>.</li> <li>● Puerta de preparación de pods soportada. Para obtener más información, véase <a href="#">Configuración del estado del pod a través de la comprobación de estado de ELB</a>.</li> </ul>	<ul style="list-style-type: none"> <li>● Se ha mejorado la fiabilidad de la tabla de flujo cuando la red subyacente es anormal.</li> <li>● Se ha mejorado la estabilidad del sistema operativo con una versión posterior del núcleo en escenarios de reinicio, como un apagado inesperado.</li> <li>● Métricas optimizadas de la GPU/NPU de cAdvisor.</li> </ul>	Se han corregido algunos problemas de seguridad.

Versión del parche del clúster de CCE	Versión de Kubernetes	Actualizaciones de funciones	Optimización	Fijación de vulnerabilidades
v1.21.7-r0	<a href="#">v1.21.14</a>	<ul style="list-style-type: none"> <li>● Soporta SFS 3.0.</li> <li>● Soporta la detección de fallas del dispositivo y el aislamiento de los nodos de la GPU.</li> <li>● Admite grupos de seguridad personalizados por clúster.</li> <li>● Los ENIs Trunkport a nivel de nodo pueden estar preenlazados para los clústeres de CCE Turbo.</li> <li>● Puede recopilar logs de plano de control.</li> </ul>	Optimiza la estabilidad de los Services/ingresos de LoadBalancer en escenarios con un gran número de conexiones.	Se han corregido algunos problemas de seguridad y las siguientes vulnerabilidades de CVE: <ul style="list-style-type: none"> <li>● <a href="#">CVE-2022-3294</a></li> <li>● <a href="#">CVE-2022-3162</a></li> <li>● <a href="#">CVE-2022-3172</a></li> </ul>
v1.21.5-r10	<a href="#">v1.21.7</a>	Soporta la creación de nodos de Arm.	<ul style="list-style-type: none"> <li>● Mejora la estabilidad del enlace de UDP después de reiniciar el nodo de BMS que ejecuta EulerOS 2.3.</li> <li>● Optimiza la estabilidad de la asignación de red cuando la red entre nodos en el plano de control se desconecta intermitentemente.</li> <li>● Soporta el endurecimiento de vulnerabilidades de OVS.</li> </ul>	Se han corregido algunos problemas de seguridad.

Versión del parche del clúster de CCE	Versión de Kubernetes	Actualizaciones de funciones	Optimización	Fijación de vulnerabilidades
v1.21.5-r0	<a href="#">v1.21.7</a>	-	<ul style="list-style-type: none"> <li>● Mejora la estabilidad del modo de red de túnel contenedor durante la actualización del clúster.</li> <li>● Mejora la capacidad de restricciones de propagación de la topología de pod.</li> <li>● Mejora la estabilidad de la liberación de enlace cuando se apaga un nodo en el plano de control del clúster.</li> <li>● Mejora la estabilidad del montaje simultáneo del volumen de almacenamiento en los pods de los nodos.</li> </ul>	Se han corregido algunos problemas de seguridad.
v1.21.4-r10	<a href="#">v1.21.7</a>	-	Mejora la estabilidad de NetworkManager en EulerOS 2.9.	Se han corregido algunos problemas de seguridad.
v1.21.4-r0	<a href="#">v1.21.7</a>	-	<ul style="list-style-type: none"> <li>● Mejora la estabilidad de la conmutación de controladores de ENI en los contenedores de Kata.</li> <li>● Mejora la estabilidad del acceso al Servicio LoadBalancer durante la actualización de la carga de trabajo y el ajuste de nodos.</li> </ul>	Se han corregido algunos problemas de seguridad.
v1.21.3-r10	<a href="#">v1.21.7</a>	-	Mejora la estabilidad de la conmutación de controladores de ENI en los contenedores de Kata.	Se han corregido algunos problemas de seguridad.

Versión del parche del clúster de CCE	Versión de Kubernetes	Actualizaciones de funciones	Optimización	Fijación de vulnerabilidades
v1.21.2-r10	<a href="#">v1.21.7</a>	-	Mejora la estabilidad del escenario en el que los servicios se interconectan con los balanceadores de carga.	Se han corregido algunos problemas de seguridad.
v1.21.1-r1	<a href="#">v1.21.7</a>	-	La red de contenedor admite la coincidencia amplia del núcleo.	Se han corregido algunos problemas de seguridad.
v1.21.1-r0	<a href="#">v1.21.7</a>	El clúster de CCE v1.21 se publica por primera vez. Para obtener más información, consulte <a href="#">Notas del lanzamiento de CCE Kubernetes 1.21</a> .	-	-

## Versión 1.19

**Tabla 2-13** Notas de la versión del parche v1.19

Versión del parche del clúster de CCE	Versión de Kubernetes	Actualizaciones de funciones	Optimización	Fijación de vulnerabilidades
v1.19.16-r20	<a href="#">v1.19.16</a>	<ul style="list-style-type: none"> <li>● Los servicios y las entradas soportan los balanceadores de carga dedicados que han estado vinculados a los GEIP.</li> <li>● Los ENIs de CCE Turbo admiten direcciones IP fijas.</li> <li>● CCE Turbo ENIs soporta la creación y encuadernación automática de EIPs.</li> </ul>	<ul style="list-style-type: none"> <li>● Cloud Native 2.0 Network admite la subred especificada por el espacio de nombres.</li> <li>● Optimiza la estabilidad de la extracción de imágenes cuando Docker se está ejecutando después de reiniciar el nodo.</li> <li>● Optimiza el rendimiento de asignación de ENI de los clústeres de CCE Turbo en escenarios de preenlace no completos.</li> </ul>	Se han corregido algunos problemas de seguridad.
v1.19.16-r7	<a href="#">v1.19.16</a>	-	<ul style="list-style-type: none"> <li>● Mejora la confiabilidad de Docker durante la actualización.</li> <li>● Optimiza la sincronización de tiempo de los nodos.</li> <li>● Mejora la confiabilidad de los clústeres de CCE Turbo durante los escenarios de precalentamiento.</li> </ul>	Se han corregido algunos problemas de seguridad.

Versión del parche del clúster de CCE	Versión de Kubernetes	Actualizaciones de funciones	Optimización	Fijación de vulnerabilidades
v1.19.16-r6	<a href="#">v1.19.16</a>	<ul style="list-style-type: none"> <li>● Los servicios y las entradas soportan los balanceadores de carga dedicados que han estado vinculados a los GEIP.</li> </ul>	<ul style="list-style-type: none"> <li>● Mejora la estabilidad del containerd (configurado con QoS).</li> <li>● Las entradas admiten la configuración y modificación de las políticas de reescritura de URL.</li> <li>● El uso de memoria de kube-controller-manager se optimiza cuando los recursos de CRD se actualizan con frecuencia.</li> </ul>	Se han corregido algunos problemas de seguridad.
v1.19.16-r5	<a href="#">v1.19.16</a>	<ul style="list-style-type: none"> <li>● Los puertos TCP/UDP se pueden configurar para los servicios de LoadBalancer al mismo tiempo. Para obtener más información, véase <a href="#">Configuración de la comprobación de estado para varios puertos</a>.</li> <li>● Puerta de preparación de pods soportada. Para obtener más información, véase <a href="#">Configuración del estado del pod a través de la comprobación de estado de ELB</a>.</li> </ul>	<ul style="list-style-type: none"> <li>● Se ha mejorado la fiabilidad de la tabla de flujo cuando la red subyacente es anormal.</li> <li>● Se ha mejorado la estabilidad del sistema operativo con una versión posterior del núcleo en escenarios de reinicio, como un apagado inesperado.</li> </ul>	Se han corregido algunos problemas de seguridad.

Versión del parche del clúster de CCE	Versión de Kubernetes	Actualizaciones de funciones	Optimización	Fijación de vulnerabilidades
v1.19.16-r4	<a href="#">v1.19.16</a>	<ul style="list-style-type: none"> <li>● Soporta SFS 3.0.</li> <li>● Soporta la detección de fallas del dispositivo y el aislamiento de los nodos de la GPU.</li> <li>● Admite grupos de seguridad personalizados por clúster.</li> <li>● Los clústeres de CCE Turbo admiten el enlace previo de ENI de Trunkport por nodo.</li> </ul>	<ul style="list-style-type: none"> <li>● La programación se optimiza en el escenario de mancha de nodo.</li> <li>● Mejora la estabilidad de funcionamiento a largo plazo de containerd en escenarios de unión de núcleo.</li> <li>● Optimiza la estabilidad de los Services/ingresses de LoadBalancer en escenarios con un gran número de conexiones.</li> <li>● Optimizado el uso de memoria de kube-apiserver cuando los recursos CRD se actualizan con frecuencia.</li> </ul>	<p>Se han corregido algunos problemas de seguridad y las siguientes vulnerabilidades de CVE:</p> <ul style="list-style-type: none"> <li>● <a href="#">CVE-2022-3294</a></li> <li>● <a href="#">CVE-2022-3162</a></li> <li>● <a href="#">CVE-2022-3172</a></li> </ul>

Versión del parche del clúster de CCE	Versión de Kubernetes	Actualizaciones de funciones	Optimización	Fijación de vulnerabilidades
v1.19.16-r3	<a href="#">v1.19.16</a>	-	<ul style="list-style-type: none"> <li>● El parámetro de inicio image-pull-progress-deadline se puede reservar durante la actualización.</li> <li>● Los clústeres de CCE Turbo admiten la preencuadración de ENI personalizada.</li> <li>● Para los nodos de BMS que ejecutan EulerOS 2.3, la estabilidad del enlace de UDP se optimiza después de reiniciar el nodo.</li> <li>● Soluciona el problema de que la asignación de red es inconsistente causada por la desconexión intermitente de la red del túnel entre los nodos maestros.</li> <li>● Mejora la estabilidad de los clústeres (con el modelo de red de túnel) cuando la red del nodo del plano de control está desconectada intermitentemente.</li> </ul>	Se han corregido algunos problemas de seguridad.
v1.19.16-r2	<a href="#">v1.19.16</a>	-	<ul style="list-style-type: none"> <li>● Mejora la estabilidad de la liberación de enlace cuando se apaga un nodo en el plano de control del clúster.</li> <li>● Mejora la estabilidad del montaje simultáneo del volumen de almacenamiento en los pods de los nodos.</li> </ul>	Se han corregido algunos problemas de seguridad.

Versión del parche del clúster de CCE	Versión de Kubernetes	Actualizaciones de funciones	Optimización	Fijación de vulnerabilidades
v1.19.16-r1	<a href="#">v1.19.16</a>	-	Mejora la estabilidad de NetworkManager en EulerOS 2.9.	Se han corregido algunos problemas de seguridad.
v1.19.16-r0	<a href="#">v1.19.16</a>	-	Mejora la estabilidad de actualización de los Services de LoadBalancer cuando se actualizan las cargas de trabajo y se amplían los nodos.	Se han corregido algunos problemas de seguridad y las siguientes vulnerabilidades de CVE: <ul style="list-style-type: none"> <li>● <a href="#">CVE-2021-25741</a></li> <li>● <a href="#">CVE-2021-25737</a></li> </ul>
v1.19.10-r0	<a href="#">v1.19.10</a>	El clúster de CCE v1.19 se publica por primera vez. Para obtener más información, consulte <a href="#">Kubernetes 1.19 (EOM) Release Notes</a> .	-	-

## 2.2 Compra de un clúster de CCE Turbo

Los clústeres CCE Turbo se ejecutan en una infraestructura nativa en la nube que ofrece sinergia de software y hardware para admitir redes de transferencia, alta seguridad y confiabilidad y programación inteligente.

Los clústeres de CCE Turbo se combinan con el modelo Cloud Native Network 2.0 para el despliegue de contenedor de alto rendimiento y de gran escala. A los contenedores se les asignan direcciones IP desde el bloque CIDR de VPC. Los contenedores y los nodos pueden pertenecer a diferentes subredes. Las solicitudes de acceso desde redes externas en una VPC

se pueden enrutar directamente a direcciones IP de contenedor, lo que mejora enormemente el rendimiento de la red. **Se recomienda** que pasa por [Cloud Native Network 2.0](#) para comprender las características y la planificación de red de cada bloque CIDR de Cloud Native Network 2.0.

## Restricciones

- Durante la creación del nodo, los paquetes de software se descargan de OBS usando el nombre de dominio. Debe utilizar un servidor de DNS privado para resolver el nombre de dominio OBS y configurar la dirección del servidor de DNS de la subred donde reside el nodo con una [dirección del servidor DNS privado](#). Cuando se crea una subred, el servidor de DNS privado se utiliza de forma predeterminada. Si cambia el DNS de subred, asegúrese de que el servidor de DNS en uso puede resolver el nombre de dominio de OBS.
- Puede crear un máximo de 50 clústeres en una sola región. Si se requieren más clústeres, puede hacer clic en la [consola](#) para aumentar su cuota. Para obtener más información sobre la cuota, consulte la sección [Cuotas](#).
- Los clústeres de CCE Turbo solo admiten Cloud Native Network 2.0. Para obtener más información acerca de este modelo de red, consulte [Cloud Native Network 2.0](#).
- Después de crear un clúster, no se pueden cambiar los siguientes elementos:
  - Tipo del clúster. Por ejemplo, cambie un **CCE cluster** a un **CCE Turbo cluster**.
  - Número de nodos principales en el clúster
  - AZ de un nodo principal
  - Configuración de red del clúster, como la VPC, la subred, el bloque CIDR de contenedor, el bloque CIDR de servicio, la configuración IPv6 y la configuración kube-proxy (reenvío).

## Procedimiento

**Paso 1** Inicie sesión en la consola de CCE. Elija **Clusters**. En la página mostrada, haga clic en **Buy** junto a **CCE Turbo cluster**.

**Paso 2** Especifique los parámetros del clúster.

### Ajustes básicos

- **Nombre de clúster**
- **Enterprise Project:**

Este parámetro solo se muestra para los usuarios de empresa que han habilitado la función de proyecto de empresa.

Después de seleccionar un proyecto de empresa (por ejemplo, el **default**), el clúster, los nodos del clúster, los grupos de seguridad del clúster, los grupos de seguridad de nodos y las IP elásticas (EIP) de los nodos creados automáticamente se crearán en este proyecto de empresa. Después de crear un clúster, se recomienda no modificar los proyectos de empresa de nodos, grupos de seguridad de clústeres y grupos de seguridad de nodos en el clúster.

Los proyectos empresariales facilitan la gestión a nivel de proyectos y el agrupamiento de los recursos y usuarios en la nube. Para obtener más información, consulte [Gestión de empresas](#).

- **Cluster Version:** Seleccione la versión de Kubernetes utilizada por el clúster.

- **Cluster Scale:** Seleccione el número máximo de nodos que puede gestionar el clúster. Una vez completada la creación, solo se admite expansión, pero no reducción.
- **HA:** modo de distribución de nodos principales. De forma predeterminada, los nodos principales se distribuyen aleatoriamente en diferentes AZ para mejorar las capacidades de DR.

También puede ampliar la configuración avanzada y personalizar el modo de distribución del nodo principal. Se soportan los siguientes modos:

- **Host:** Los nodos principales se crean en diferentes hosts en la misma AZ.
- **Custom:** Puede determinar la ubicación de cada nodo principal.

### Ajustes de redes

La configuración de la red del clúster abarca los nodos, los contenedores y los Services. Para obtener más información acerca de la red de clústeres y los modelos de red de contenedor, consulte la sección [Descripción general](#).

- **Network Model:** Los clústeres Turbo de CCE solo admiten **Cloud Native Network 2.0**. Para obtener más información, véase [Cloud Native Network 2.0](#).
- **VPC:** Seleccione la VPC a la que pertenece el clúster. Si no hay ninguna VPC disponible, haga clic en **Create VPC** para crear una. El valor no se puede cambiar después de la creación.
- **Master Node Subnet:** Seleccione la subred donde se despliega el nodo principal. Si no hay ninguna subred disponible, haga clic en **Create Subnet** para crear una. Un nodo principal requiere al menos cuatro direcciones IP, que no se pueden cambiar después de la creación.
- **Pod Subnet:** Seleccione la subred donde se encuentra el contenedor. Si no hay ninguna subred disponible, haga clic en **Create Subnet** para crear una. La subred pod determina el número máximo de contenedores en el clúster. Puede agregar subredes de pod después de crear el clúster.
- **Service CIDR Block:** Bloque CIDR para los [Services](#) utilizados por contenedores en el mismo clúster para acceder entre sí. El valor determina el número máximo de Services que puede crear. El valor no se puede cambiar después de la creación.

### Configuración avanzada

- **Request Forwarding:** Se admiten los modos IPVS e iptables. Para obtener más información, véase [Comparación de iptables e IPVS](#).
- **CPU Manager:** Para más información, véase [Política de CPU](#).
- **Resource Tag:**

Puede agregar etiquetas de recursos para clasificar recursos.

Puede crear **etiquetas predefinidas** en Tag Management Service (TMS). Las etiquetas predefinidas son visibles para todos los recursos de servicio que admiten la función de etiquetado. Puede utilizar las etiquetas predefinidas para mejorar la creación de etiquetas y la eficiencia de la migración de recursos. Para obtener más información, consulte [Creación de etiquetas predefinidas](#).

Especificaciones de clave

- Es un campo obligatorio. Contiene de 1 a 128 caracteres de un solo byte.
- No introduzca etiquetas que comiencen por `_sys_`, que son etiquetas del sistema.
- Puede contener letras de UTF-8 (incluidos los caracteres chinos), dígitos, espacios y los siguientes caracteres: `_ . : / = + - @`

Expresión regular recomendada: `^((?!_sys_)[\p{L}\p{Z}\p{N}_:\V=+\\-@]*)$`

Especificaciones de valor

- Puede contener hasta 255 caracteres.
- Puede contener letras de UTF-8 (incluidos los caracteres chinos), dígitos, espacios y los siguientes caracteres: `_ . : / = + - @`

Expresión regular recomendada: `^[\\p{L}\\p{Z}\\p{N}_:\V=+\\-@]*$`

- El valor puede ser vacío o nulo.
- El valor de una etiqueta predefinida no puede estar vacío o nulo.
- **Default Node Security Group:** Puede utilizar el grupo de seguridad generado automáticamente por CCE o seleccionar uno existente.

---

### AVISO

El grupo de seguridad de nodo predeterminado necesita permitir el acceso desde ciertos puertos para garantizar una comunicación normal. De lo contrario, no se puede crear el nodo. Para obtener más información, consulte [Configuración de reglas de grupo de seguridad de clúster de CCE](#).

- **Certificate Authentication:**
    - **Default:** El modo de autenticación basado en X509 está habilitado de forma predeterminada. X509 es un formato de certificado de uso común.
    - **Custom:** El clúster puede identificar a los usuarios basándose en el encabezado del cuerpo de la solicitud para la autenticación.
- Necesita cargar su **CA root certificate**, **client certificate** y **private key** del certificado de cliente.

---

### ⚠ ATENCIÓN

- Cargue un archivo **menos de 1 MB**. El certificado de CA y el certificado de cliente pueden estar en formato **.crt** o **.cer**. La clave privada del certificado de cliente solo se puede cargar **unencrypted**.
  - El período de validez del certificado de cliente debe ser superior a cinco años.
  - El certificado de CA cargado se utiliza tanto para el proxy de autenticación como para la configuración de la capa de agregación kube-apiserver. **Si el certificado no es válido, no se puede crear el clúster.**
  - A partir de la v1.25, Kubernetes ya no admite la autenticación de certificados generada mediante el algoritmo SHA1WithRSA o ECDSAWithSHA1. Se recomienda utilizar el algoritmo SHA256.
- 
- **Descripción** La descripción no puede exceder los 200 caracteres.

**Paso 3** Haga clic en **Next: Add-on Configuration**.

**Domain Name Resolution:** El complemento de **coredns** se instala de forma predeterminada para resolver nombres de dominio y conectarse al servidor de DNS en la nube.

**Container Storage:** El complemento [everest](#) se instala de forma predeterminada para proporcionar almacenamiento de contenedor basado en CSI y conectarse a servicios de almacenamiento en la nube como EVS.

#### Service Logs

- Uso de ICAgent:

Un recopilador de logs proporcionado por Application Operations Management (AOM), que informa de logs a AOM y Log Tank Service (LTS) de acuerdo con las reglas de recopilación de registros configuradas.

Puede recopilar logs stdout según sea necesario.

**Overload Control:** Si está habilitado, las solicitudes simultáneas se controlan dinámicamente en función de la presión de recursos de los nodos maestros para mantenerlas disponibles y el clúster.

**Paso 4** Después de configurar los parámetros, haga clic en **Next: Confirm**. Revise la configuración y seleccione un modo de pago y la duración requerida.

- **Pay-per-use:** Haga clic en **Submit**.
- **Yearly/Monthly:** Haga clic en **Pay Now**. En la página que se muestra, haga clic en **Pay**.

Se tarda entre 6 y 10 minutos en crear un clúster. Puede hacer clic en **Back to Cluster List** para realizar otras operaciones en el clúster o hacer clic en **Go to Cluster Events** para ver los detalles del clúster.

---Fin

## Operaciones relacionadas

- Usar kubectl para conectarse al clúster: [Conexión a un clúster con kubectl](#)
- Agregar nodos al clúster. Para obtener más información, véase [Creación de un nodo](#).

## 2.3 Compra de un clúster de CCE

En la consola de CCE, puede crear fácilmente los clústeres de Kubernetes. Kubernetes puede gestionar los clústeres de contenedor a escala. Un clúster gestiona un grupo de recursos de nodo.

En CCE, puede crear un clúster de CCE para gestionar las VM y los servidores de metal puro (BMS) como nodos, y los nodos heterogéneos habilitados para GPU y NPU. Mediante el uso de modelos de red de alto rendimiento, los clústeres híbridos proporcionan un entorno de tiempo de ejecución estable, seguro y de múltiples escenarios para contenedores.

## Restricciones

- Durante la creación del nodo, los paquetes de software se descargan de OBS usando el nombre de dominio. Debe utilizar un servidor de DNS privado para resolver el nombre de dominio OBS y configurar la dirección del servidor de DNS de la subred donde reside el nodo con una [dirección del servidor DNS privado](#). Cuando se crea una subred, el servidor de DNS privado se utiliza de forma predeterminada. Si cambia el DNS de subred, asegúrese de que el servidor de DNS en uso puede resolver el nombre de dominio de OBS.
- Puede crear un máximo de 50 clústeres en una sola región. Si se requieren más clústeres, puede hacer clic en [la consola](#) para aumentar su cuota.

- Después de crear un clúster, no se pueden cambiar los siguientes elementos:
  - Tipo de clúster
  - Número de nodos principales en el clúster
  - AZ de un nodo principal
  - Configuración de red del clúster, como la VPC, la subred, el bloque CIDR de contenedor, el bloque CIDR de servicio, la configuración IPv6 y la configuración kube-proxy (reenvío)
  - Modelo de red. Por ejemplo, cambie **Tunnel network** a **VPC network**.

## Procedimiento

**Paso 1** Inicie sesión en la consola de CCE. Elija **Clusters**. En la página mostrada, haga clic en **Buy** junto a **CCE cluster**.

**Paso 2** Establezca los parámetros del clúster.

### Ajustes básicos

- **Cluster Name**
- **Enterprise Project:**

Este parámetro solo se muestra para los usuarios de empresa que han habilitado la función de proyecto de empresa.

Después de seleccionar un proyecto de empresa (por ejemplo, el **default**), el clúster, los nodos del clúster, los grupos de seguridad del clúster, los grupos de seguridad de nodos y las IP elásticas (EIP) de los nodos creados automáticamente se crearán en este proyecto de empresa. Después de crear un clúster, se recomienda no modificar los proyectos de empresa de nodos, grupos de seguridad de clústeres y grupos de seguridad de nodos en el clúster.

Un proyecto empresarial facilita la gestión a nivel de proyectos y el agrupamiento de los recursos y usuarios en la nube. Para obtener más información, consulte [Gestión de empresas](#).

- **Cluster Version:** Seleccione la versión de Kubernetes utilizada por el clúster.
- **Cluster Scale:** Número máximo de nodos que puede gestionar el clúster. Una vez completada la creación, solo se admite expansión, pero no reducción.
- **HA:** modo de distribución de nodos principales. De forma predeterminada, los nodos principales se distribuyen aleatoriamente en diferentes AZ para mejorar las capacidades de DR.

También puede ampliar la configuración avanzada y personalizar el modo de distribución del nodo principal. Se admiten los dos modos siguientes:

- **Random:** Los nodos principales se crean en diferentes AZ para DR.
- **Custom:** Puede determinar la ubicación de cada nodo principal.
  - **Host:** Los nodos principales se crean en diferentes hosts en la misma AZ.
  - **Custom:** Puede determinar la ubicación de cada nodo principal.

### Ajustes de redes

La configuración de la red del clúster abarca los nodos, los contenedores y los Services. Para obtener más información acerca de la red de clústeres y los modelos de red de contenedor, consulte la sección [Descripción general](#).

- **Network Model:** Los clústeres de CCE admiten modelos **VPC network** y **tunnel network**. Para más detalles, véase [Red de VPC](#) y [Red de túneles de contenedores](#).
- **VPC:** Seleccione la VPC a la que pertenece el clúster. Si no hay ninguna VPC disponible, haga clic en **Create VPC** para crear una. El valor no se puede cambiar después de la creación. La pila dual IPv4/IPv6 está disponible para clústeres de CCE de v1.15 y posteriores y estará disponible generalmente para clústeres de v1.23. Para obtener más información, consulte [Creación de un clúster de doble pila IPv4/IPv6 en CCE](#).
- **Master Node Subnet:** Seleccione la subred donde se despliega el nodo principal. Si no hay ninguna subred disponible, haga clic en **Create Subnet** para crear una. La subred no se puede cambiar después de la creación.
- **Container CIDR Block:** Establezca el bloque CIDR usado por contenedores. Se pueden agregar varios bloques CIDR de contenedor al modelo de red de VPC después de crear un clúster.
- **Service CIDR Block:** Bloque CIDR para los Services utilizados por contenedores en el mismo clúster para acceder entre sí. El valor determina el número máximo de Services que puede crear. El valor no se puede cambiar después de la creación.

### Configuración avanzada

- **Request Forwarding:** Se admiten los modos IPVS e iptables. Para obtener más información, véase [Comparación de iptables e IPVS](#).
- **CPU Manager:** Para más información, véase [Política de CPU](#).
- **Resource Tag:**

Puede agregar etiquetas de recursos para clasificar recursos.

Puede crear etiquetas predefinidas en Tag Management Service (TMS). Las etiquetas predefinidas son visibles para todos los recursos de servicio que admiten la función de etiquetado. Las etiquetas predefinidas se pueden usar para mejorar la creación de etiquetas y la eficacia de la migración. Para obtener más información, consulte [Creación de etiquetas predefinidas](#).

Especificaciones de clave

- No puede estar vacío. Contiene de 1 a 128 caracteres de un solo byte.
- No introduzca etiquetas que comiencen por `_sys_`, que son etiquetas del sistema.
- Puede contener letras de UTF-8 (incluidos los caracteres chinos), dígitos, espacios y los siguientes caracteres: `_ . : / = + - @`

Expresión regular recomendada: `^(?!_sys_)[\p{L}\p{Z}\p{N}_.:\/=+\\-@]*$`

Especificaciones de valor

- Puede contener hasta 255 caracteres.
- Puede contener letras de UTF-8 (incluidos los caracteres chinos), dígitos, espacios y los siguientes caracteres: `_ . : / = + - @`

Expresión regular recomendada: `^([\p{L}\p{Z}\p{N}_.:\/=+\\-@]*)$`

- El valor puede ser vacío o nulo.
- El valor de una etiqueta predefinida no puede estar vacío o nulo.

- **Default Node Security Group:** Puede utilizar el grupo de seguridad generado automáticamente por CCE o seleccionar uno existente.

### AVISO

El grupo de seguridad de nodo predeterminado necesita permitir el acceso desde ciertos puertos para garantizar una comunicación normal. De lo contrario, no se puede crear el nodo. Para obtener más información, consulte [Configuración de reglas de grupo de seguridad de clúster de CCE](#).

- **Certificate Authentication:**

- **Default:** El modo de autenticación basado en X509 está habilitado de forma predeterminada. X509 es un formato de certificado de uso común.
- **Custom:** El clúster puede identificar a los usuarios basándose en el encabezado del cuerpo de la solicitud para la autenticación.

Necesita cargar su **CA root certificate**, **client certificate** y **private key** del certificado de cliente.

### ⚠ ATENCIÓN

- Cargue un archivo **menos de 1 MB**. El certificado de CA y el certificado de cliente pueden estar en formato **.crt** o **.cer**. La clave privada del certificado de cliente solo se puede cargar **unencrypted**.
- El período de validez del certificado de cliente debe ser superior a cinco años.
- El certificado de CA cargado se utiliza tanto para el proxy de autenticación como para la configuración de la capa de agregación kube-apiserver. **Si el certificado no es válido, no se puede crear el clúster.**
- A partir de la v1.25, Kubernetes ya no admite la autenticación de certificados generada mediante el algoritmo SHA1WithRSA o ECDSAWithSHA1. Se recomienda utilizar el algoritmo SHA256.

- Otros parámetros (haciendo clic en **Manage** en la tarjeta del clúster)

**Overload Control:** Si está habilitado, las solicitudes simultáneas se controlan dinámicamente en función de la presión de recursos de los nodos maestros para mantenerlas disponibles y el clúster.

- **Descripción** La descripción no puede exceder los 200 caracteres.

### Paso 3 Haga clic en **Next: Add-on Configuration**.

**Domain Name Resolution:** El complemento de **coredns** se instala de forma predeterminada para resolver nombres de dominio y conectarse al servidor de DNS en la nube.

**Container Storage:** El complemento **everest** se instala de forma predeterminada para proporcionar almacenamiento de contenedor basado en CSI y conectarse a servicios de almacenamiento en la nube como EVS.

### Service Logs

- Uso de ICAgent:

Un recopilador de logs proporcionado por Application Operations Management (AOM), que informa de logs a AOM y Log Tank Service (LTS) de acuerdo con las reglas de recopilación de registros configuradas.

Puede recopilar logs stdout según sea necesario.

**Overload Control:** Si está habilitado, las solicitudes simultáneas se controlan dinámicamente en función de la presión de recursos de los nodos maestros para mantenerlas disponibles y el clúster.

**Paso 4** Después de especificar los parámetros, haga clic en **Next: Confirm** Ajustar. Se muestra la lista de recursos del clúster. Confirme la información y haga clic en **Submit**.

Se tarda entre 6 y 10 minutos en crear un clúster. Puede hacer clic en **Back to Cluster List** para realizar otras operaciones en el clúster o hacer clic en **Go to Cluster Events** para ver los detalles del clúster.

---Fin

## Operaciones relacionadas

- Después de crear un clúster, puede usar la herramienta de línea de comandos (CLI) de Kubernetes `kubectl` para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).
- Agregar nodos al clúster. Para obtener más información, véase [Creación de un nodo](#).
- Cree un clúster de doble pila IPv4/IPv6. Para obtener más información, consulte [Creación de un clúster de doble pila IPv4/IPv6 en CCE](#).
- Conéctese a varios clústeres mediante `kubectl`. Para obtener más información, consulte [Conexión a varios clústeres con kubectl](#).

## 2.4 Conexión de clústeres

### 2.4.1 Conexión a un clúster con kubectl

#### Escenario

Esta sección utiliza un clúster de CCE como ejemplo para describir cómo conectarse a un clúster de CCE con `kubectl`.

#### Descripción del permiso

Cuando se accede a un clúster mediante `kubectl`, CCE utiliza el archivo `kubeconfig.json` generado en el clúster para la autenticación. Este archivo contiene información del usuario, basada en la cual CCE determina qué recursos de Kubernetes puede acceder `kubectl`. Los permisos registrados en un archivo `kubeconfig.json` varían de usuario a usuario.

Para obtener más información acerca de los permisos de usuario, consulte [Cluster Permissions \(basados en IAM\) y Namespace Permissions \(basados en Kubernetes RBAC\)](#).

#### Uso de kubectl

Para conectarse a un clúster de Kubernetes desde un PC, puede usar `kubectl`, una herramienta de línea de comandos de Kubernetes. Puede iniciar sesión en la consola de CCE, hacer clic en el nombre del clúster que se va a conectar y ver la dirección de acceso y el procedimiento de conexión `kubectl` en la página de detalles del clúster como se muestra en la sección [Figura 2-3](#).

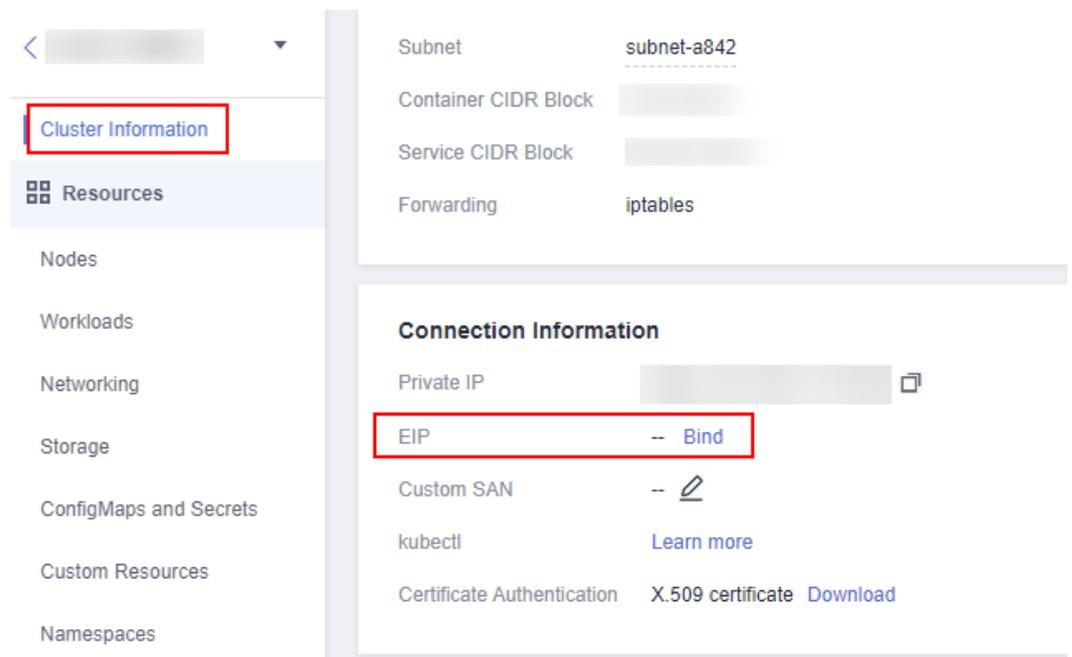
CCE le permite acceder a un clúster con un **VPC network** o un **public network**.

- **Intra-VPC access:** El cliente que accede al clúster debe estar en la misma VPC que el clúster.
- **Public access:** El cliente que accede al clúster debe poder acceder a las redes públicas y el clúster se ha vinculado con una IP de red pública.

**AVISO**

Para vincular una IP pública (EIP) al clúster, vaya a la página de detalles del clúster y haga clic en **Bind** junto a **EIP** en el panel **Connection Information**, como se muestra en **Figura 2-3**. En un clúster con una EIP vinculada, kube-apiserver estará expuesto a redes públicas y puede ser atacado. Se recomienda configurar Advanced Anti-DDoS (AAD) para la EIP del nodo donde reside kube-apiserver.

**Figura 2-3** Información de conexión de clúster



Descargue kubectl y el archivo de configuración. Copie el archivo a su cliente y configure kubectl. Una vez completada la configuración, puede acceder a sus clústeres de Kubernetes. Procedimiento:

**Paso 1 Descargar kubectl.**

Prepare un equipo que pueda acceder a la red pública e instale kubectl en modo de CLI. Puede ejecutar el comando **kubectl version** para comprobar si kubectl ha sido instalado. Si se ha instalado kubectl, omite este paso.

Esta sección utiliza el entorno de Linux como ejemplo para describir cómo instalar y configurar kubectl. Para más detalles, véase **Instalación de kubectl**.

1. Inicie sesión en su cliente y descargue kubectl.

```
cd /home
curl -LO https://dl.k8s.io/release/{v1.25.0}/bin/linux/amd64/kubectl
```

`{v1.25.0}` especifica el número de versión. Reemplácelo según sea necesario.

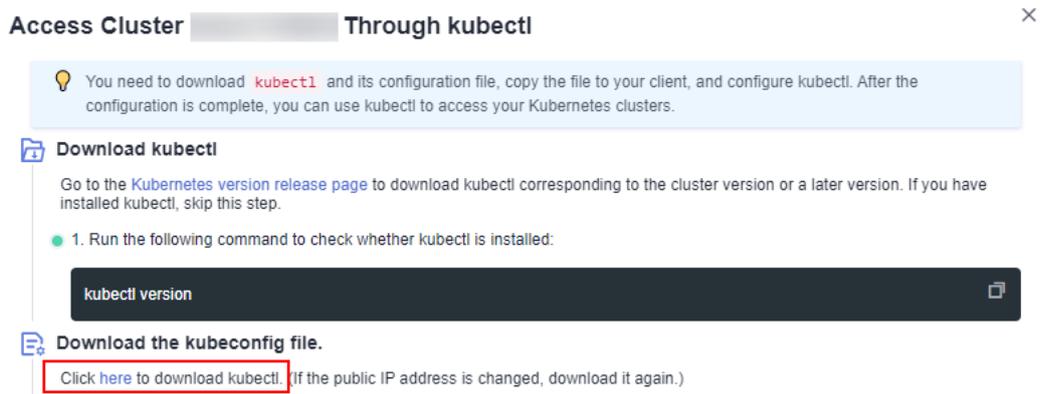
2. Instale kubectl.

```
chmod +x kubectl
mv -f kubectl /usr/local/bin
```

**Paso 2 Obtener el archivo de configuración de kubectl (kubeconfig).**

En el panel **Connection Information** de la página de detalles del clúster, haga clic en **Learn more** junto a **kubectl**. En la ventana que se muestra, descargue el archivo de configuración.

**Figura 2-4** Descargar el archivo de configuración



**NOTA**

- Se utiliza el archivo de configuración de kubectl **kubeconfig.json** para la autenticación del clúster. Si se filtra el archivo, los clústeres pueden ser atacados.
- De forma predeterminada, la autenticación bidireccional está deshabilitada para los nombres de dominio en el clúster actual. Puede ejecutar el comando **kubectl config use-context externalTLSVerify** para habilitar la autenticación bidireccional. Para obtener más información, consulte **Autenticación bidireccional para nombres de dominio**. Para un clúster que se ha vinculado a una EIP, si la autenticación falla (x509: certificado es válido) cuando se utiliza la autenticación bidireccional, debe vincular la EIP de nuevo y descargar **kubeconfig.json** de nuevo.
- Los permisos de Kubernetes asignados por el archivo de configuración descargado por los usuarios de IAM son los mismos que los asignados a los usuarios de IAM en la consola de CCE.
- Si la variable de entorno KUBECONFIG está configurada en el sistema operativo Linux, kubectl carga preferentemente la variable de entorno KUBECONFIG en lugar de **\$home/.kube/config**.

**Paso 3** Configurar kubectl.

Configurar kubectl (se utiliza un sistema operativo Linux).

1. Inicie sesión en su cliente y copie el archivo de configuración kubeconfig.json descargado en **Paso 2** al directorio **/home** de su cliente.

2. Configure el archivo de autenticación de kubectl.

```
cd /home
mkdir -p $HOME/.kube
mv -f kubeconfig.json $HOME/.kube/config
```

3. Cambie el modo de acceso kubectl basado en escenarios de servicio.

– Ejecute este comando para habilitar el acceso dentro de la VPC:

```
kubectl config use-context internal
```

– Ejecute este comando para habilitar el acceso público (se requiere la EIP):

```
kubectl config use-context external
```

- Ejecute este comando para habilitar el acceso público y la autenticación bidireccional (se requiere la EIP):

```
kubectrl config use-context externalTLSVerify
```

Para obtener más información acerca de la autenticación bidireccional del clúster, consulte [Autenticación bidireccional para nombres de dominio](#).

---Fin

## Autenticación bidireccional para nombres de dominio

Actualmente, CCE admite la autenticación bidireccional para nombres de dominio.

- La autenticación bidireccional está deshabilitada para los nombres de dominio de forma predeterminada. Puede ejecutar el comando **kubectrl config use-context externalTLSVerify** para cambiar al contexto externalTLSVerify para habilitarlo.
- Cuando una EIP está vinculada o no vinculada de un clúster, o se configura o actualiza un nombre de dominio personalizado, el certificado del servidor de clúster se agregará con la dirección de acceso al clúster más reciente (incluida la EIP vinculada al clúster y todos los nombres de dominio personalizados configurados para el clúster).
- La sincronización de clúster asincrónica tarda aproximadamente de 5 a 10 minutos. Puede ver el resultado de la sincronización en el **Synchronize Certificate** de **Operation Records**.
- Para un clúster que se ha vinculado a una EIP, si la autenticación falla (x509: certificado es válido) cuando se utiliza la autenticación bidireccional, debe vincular la EIP de nuevo y descargar **kubeconfig.json** de nuevo.
- Si no se admite la autenticación bidireccional del nombre de dominio, **kubeconfig.json** contiene el campo **"insecure-skip-tls-verify": true**, como se muestra en [Figura 2-5](#). Para utilizar la autenticación bidireccional, puede descargar de nuevo el archivo **kubeconfig.json** y habilitar la autenticación bidireccional para los nombres de dominio.

Figura 2-5 Autenticación bidireccional deshabilitada para los nombres de dominio

```
"clusters": [{  
  "name": "mycluster",  
  "cluster": {  
    "server": "https://10.100.0.52:5443",  
    "insecure-skip-tls-verify": true  
  }  
}]
```

## Preguntas frecuentes

- **Error del servidor prohibido**

Cuando usa kubectrl para crear o consultar recursos de Kubernetes, se devuelve el siguiente resultado:

```
# kubectrl get deploy Error from server (Forbidden): deployments.apps is forbidden: User "0c97ac3cb280f4d91fa7c0096739e1f8" cannot list resource "deployments" in API group "apps" in the namespace "default"
```

La causa es que el usuario no tiene los permisos para operar los recursos de Kubernetes. Para obtener más información acerca de cómo asignar permisos, consulte [Permisos de espacio de nombres \(basados en Kubernetes RBAC\)](#).

- **La conexión al servidor localhost:8080 fue rechazada**

Cuando usa kubectrl para crear o consultar recursos de Kubernetes, se devuelve el siguiente resultado:

```
The connection to the server localhost:8080 was refused - did you specify the right host or port?
```

La causa es que la autenticación del clúster no está configurada para el cliente kubectl. Para obtener más información, véase [Paso 3](#).

## Operaciones relacionadas

- [Conectarse a varios clústeres mediante kubectl.](#)
- [Configurar kubeconfig para una gestión detallada en los recursos del clúster](#)

## 2.4.2 Conexión a un clúster con CloudShell

### Escenario

En esta sección se utiliza un clúster de CCE como ejemplo para describir cómo conectarse a un clúster de CCE con CloudShell.

### Descripción del permiso

Cuando se utiliza kubectl en CloudShell, los permisos de kubectl son determinados por el usuario que inicia sesión.

### Uso de CloudShell

CloudShell es un shell web utilizado para gestionar y mantener recursos en la nube. CCE le permite usar CloudShell para conectarse a clústeres y usar kubectl de CloudShell para acceder a clústeres (haga clic en  de [Figura 2-6](#)).

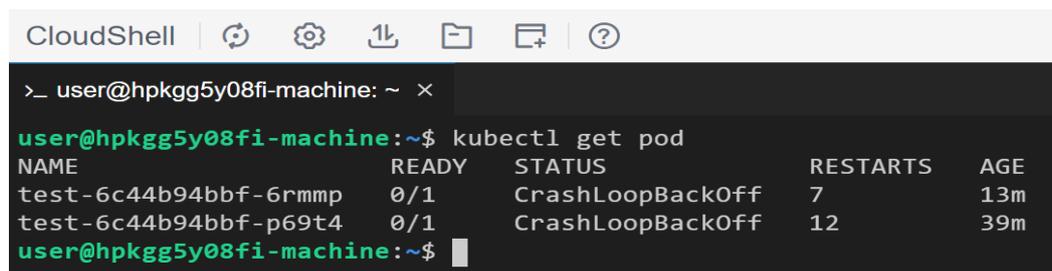
#### NOTA

- El certificado de kubectl de CloudShell es válido por un día. Puede restablecer el período de validez accediendo a CloudShell desde la consola de CCE.
- CloudShell se implementa según VPCEP. Para usar kubectl para acceder a un clúster, debe configurar el grupo de seguridad (*Cluster name-cce-control-Random number*) en el nodo principal del clúster para permitir que los siguientes bloques CIDR accedan al puerto 5443. De forma predeterminada, el puerto 5443 permite el acceso desde todos los bloques CIDR. Si tiene grupos de seguridad reforzados y no se puede acceder a ningún clúster de CloudShell, compruebe si el puerto 5443 permite el acceso desde **198.19.0.0/16**.
- CloudShell solo se puede utilizar después de que CoreDNS esté instalado en un clúster.
- Actualmente, puede usar CloudShell para iniciar sesión en contenedores solo en las regiones CN North-Beijing1, CN North-Beijing4, CN East-Shanghai1, CN East-Shanghai2, CN South-Guangzhou y CN North-Ulanqab1.

**Figura 2-6** CloudShell



Figura 2-7 Uso de kubectl en CloudShell



## 2.4.3 Conexión a un clúster con un certificado X.509

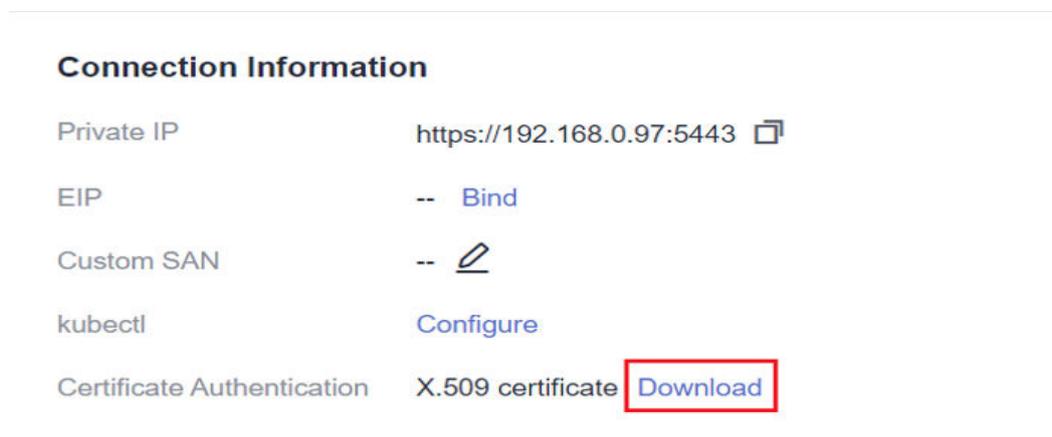
### Escenario

En esta sección se describe cómo obtener el certificado de clúster desde la consola y usarlo para acceder a los clústeres de Kubernetes.

### Procedimiento

- Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.
- Paso 2** Elija **Cluster Information** en el panel de navegación y haga clic en **Download** junto a **Authentication Mode** en el área **Connection Information**.

Figura 2-8 Descarga de un certificado de clúster



- Paso 3** En el cuadro de diálogo **Download X.509 Certificate** que se muestra, seleccione la hora de caducidad del certificado y descargue el certificado X.509 del clúster como se le solicite.

#### AVISO

- El certificado descargado contiene tres archivos: **client.key**, **client.crt** y **ca.crt**. Mantenga estos archivos seguros.
- No se requieren certificados para el acceso mutuo entre contenedores en un clúster.

----Fin

## 2.4.4 Personalización de una SAN de certificados de clúster

### Escenario

Un **Subject Alternative Name (SAN)** se puede iniciar sesión en un certificado de servidor de clúster. Por lo general, el cliente utiliza una SAN para verificar la validez del servidor en los acuerdos de enlace TLS. Específicamente, la comprobación de validez incluye si el certificado de servidor es emitido por una CA de confianza por el cliente y si la SAN en el certificado coincide con la dirección IP o el nombre de dominio DNS al que el cliente accede realmente.

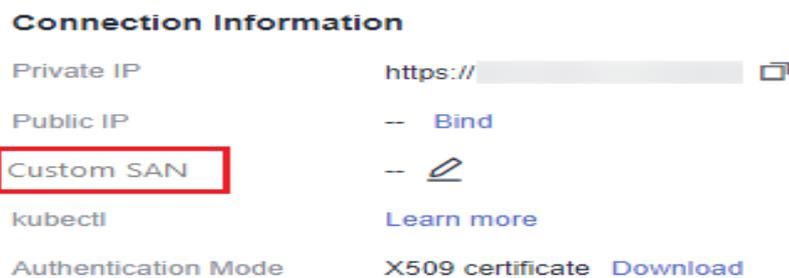
Si el cliente no puede acceder directamente a la IP privada o EIP del clúster, puede firmar la dirección IP o el nombre de dominio DNS a los que el cliente puede acceder directamente en el certificado del servidor del clúster para habilitar la autenticación bidireccional en el cliente, lo que mejora la seguridad. Los casos de uso típicos incluyen acceso de DNAT y acceso de nombre de dominio.

### Restricciones

Esta función solo está disponible para clústeres de v1.19 y versiones posteriores.

### Personalización de una SAN

- Paso 1** Inicie sesión en la consola de CCE.
- Paso 2** Haga clic en el clúster de destino en la lista de clústeres para ir a la página de detalles del clúster.
- Paso 3** En el área **Connection Information**, haga clic en  junto a **Custom SAN**. En el cuadro de diálogo que se muestra, agregue la dirección IP o el nombre de dominio y haga clic en **Save**.



#### NOTA

1. Esta operación reiniciará kube-apiserver y actualizará el archivo **kubeconfig.json** por un corto período de tiempo. No realice operaciones en el clúster durante este período.
2. Se permite un máximo de 128 nombres de dominio o direcciones IP, separados por comas (,).
3. Si un nombre de dominio personalizado necesita estar enlazado a una EIP, asegúrese de que se ha configurado una EIP.

----Fin

## Escenarios típicos de acceso a nombres de dominio

- Agregue la asignación de nombre de dominio de respuesta al especificar la dirección del nombre de dominio de DNS en la configuración del nombre de dominio del host en el cliente o al configurar `/etc/hosts` en el host del cliente.
- Utilice el acceso al nombre de dominio en la intranet. DNS le permite configurar asignaciones entre EIP de clúster y nombres de dominio personalizados. Después de actualizar una EIP, puede seguir utilizando la autenticación bidireccional y el nombre de dominio para acceder al clúster sin descargar de nuevo el archivo `kubeconfig.json`.
- Agregue los registros A en un servidor DNS autoconstruido.

## 2.4.5 Comandos comunes de kubectl

### Pasos iniciales

#### get

El comando `get` muestra uno o varios recursos de un clúster.

Este comando imprime una tabla con la información más importante sobre todos los recursos, incluidos los nodos del clúster, los pods en ejecución, las Deployments y los Services.

#### AVISO

Un clúster puede tener varios espacios de nombres. Si no se especifica ningún espacio de nombres, este comando se ejecutará con el indicador `--namespace=default`.

Por ejemplo:

Para enumerar todos los pods con información detallada:

```
kubectl get po -o wide
```

Para mostrar los pods en todos los espacios de nombres:

```
kubectl get po --all-namespaces
```

Para enumerar las etiquetas de los pods en todos los espacios de nombres:

```
kubectl get po --show-labels
```

Para listar todos los espacios de nombres del nodo:

```
kubectl get namespace
```

#### NOTA

Para listar información de otros nodos, ejecute este comando con el indicador `-s`. Para mostrar un tipo de recurso especificado, agregue el tipo de recurso a este comando, por ejemplo, `kubectl get svc`, `kubectl get nodes` y `kubectl get deploy`.

Para enumerar un pod con un nombre especificado en el formato de salida YAML:

```
kubectl get po <podname> -o yaml
```

Para enumerar un pod con un nombre especificado en formato de salida JSON:

```
kubectl get po <podname> -o json  
kubectl get po rc-nginx-2-btv4j -o=custom-columns=LABELS:.metadata.labels.app
```

### NOTA

**LABELS** indica una lista separada por comas de títulos de columna definidos por el usuario.  
**metadata.labels.app** indica los datos a listar en formato de salida YAML o JSON.

### **create**

El comando **create** crea un recurso de clúster a partir de un archivo o entrada.

Si ya hay un descriptor de recurso (un archivo YAML o JSON), puede crear el recurso desde el archivo ejecutando el siguiente comando:

```
kubectl create -f filename
```

### **expose**

El comando **expose** expone un recurso como un nuevo servicio de Kubernetes. Los recursos posibles incluyen un pod, Service y Deployment.

```
kubectl expose deployment deployname --port=81 --type=NodePort --target-port=80 --name=service-name
```

### NOTA

En el comando anterior, **--port** indica el puerto expuesto por el Service, **--type** indica el tipo de Service y **--target-port** indica el puerto del pod que respalda el Service. Visitar *ClusterIP:Port* le permite acceder a las aplicaciones del clúster.

### **run**

Por ejemplo:

Para ejecutar una imagen concreta en el clúster:

```
kubectl run deployname --image=nginx:latest
```

Para ejecutar una imagen concreta con un comando especificado:

```
kubectl run deployname -image=busybox --command -- ping baidu.com
```

### **set**

El comando **set** configura los recursos de objeto.

Por ejemplo:

Para cambiar la imagen de un despliegue con el nombre especificado en **deployname** a la imagen 1.0:

```
kubectl set image deploy deployname containername=containername:1.0
```

### **edit**

El comando **edit** edita un recurso del editor predeterminado.

Por ejemplo:

Para actualizar un pod:

```
kubectl edit po po-nginx-btv4j
```

El comando de ejemplo produce el mismo efecto que el siguiente comando:

```
kubectl get po po-nginx-btv4j -o yaml >> /tmp/nginx-tmp.yaml  
vim /tmp/nginx-tmp.yaml
```

```
/*do some changes here */  
kubectl replace -f /tmp/nginx-tmp.yaml
```

### **explain**

El comando **explain** visualiza documentos o documentos de referencia.

Por ejemplo:

Para obtener la documentación de los pods:

```
kubectl explain pod
```

### **delete**

El comando **delete** elimina los recursos por nombre de recurso o etiqueta.

Por ejemplo:

Para eliminar un pod con un retraso mínimo:

```
kubectl delete po podname --now  
kubectl delete -f nginx.yaml  
kubectl delete deployment deployname
```

## Comandos de despliegue

### **rolling-update\***

**rolling-update** es un comando muy importante. Actualiza un servicio en ejecución sin tiempo de inactividad. Los pods son reemplazados incrementalmente por otros nuevos. Un pod se actualiza a la vez. El pod antiguo se elimina solo después de que el nuevo pod esté activado. Los pods nuevos deben ser distintos de los pods antiguos por nombre, versión y etiqueta. De lo contrario, se informará de un mensaje de error.

```
kubectl rolling-update poname -f newfilename  
kubectl rolling-update poname -image=image:v2
```

Si se produce algún problema durante la actualización continua, ejecute el comando con el indicador **-rollback** para cancelar la actualización continua y volver al pod anterior.

```
kubectl rolling-update poname -rollback
```

### **rollout**

El comando **rollout** gestiona el lanzamiento de un recurso.

Por ejemplo:

Para comprobar el estado de despliegue de un lanzamiento concreto:

```
kubectl rollout status deployment/deployname
```

Para ver el historial de despliegue de un lanzamiento concreto:

```
kubectl rollout history deployment/deployname
```

Para volver al despliegue anterior: (de forma predeterminada, un recurso se revierte a la versión anterior)

```
kubectl rollout undo deployment/test-nginx
```

### **scale**

El comando **scale** establece un nuevo tamaño para un recurso ajustando el número de réplicas de recursos.

```
kubectl scale deployment deployname --replicas=newnumber
```

### autoscale

El comando **autoscale** selecciona y establece automáticamente el número de pods. Este comando especifica el rango para el número de réplicas de pod mantenidas por un controlador de replicación. Si hay demasiados pods, el controlador de replicación termina los pods adicionales. Si hay muy pocos, el controlador de replicación inicia más pods.

```
kubectl autoscale deployment deployname --min=minnumber --max=maxnumber
```

## Comandos de gestión de clústeres

### cordons, drains, uncordons\*

Si un nodo que se va a actualizar está ejecutando muchos pods o ya está inactivo, realice los siguientes pasos para preparar el nodo para el mantenimiento:

- Paso 1** Ejecute el comando **cordons** para marcar un nodo como no programado. Esto significa que los nuevos pods no se programarán en el nodo.

```
kubectl cordon nodename
```

Nota: En CCE, el **nodename** indica la dirección IP de red privada de un nodo.

- Paso 2** Ejecute el comando **drains** para migrar sin problemas los pods en ejecución desde el nodo a otro nodo.

```
kubectl drain nodename --ignore-daemonsets --ignore-emptydir
```

**ignore-emptydir** ignora los pods que usan emptyDirs.

- Paso 3** Realice operaciones de mantenimiento en el nodo, como actualizar el núcleo y actualizar Docker.

- Paso 4** Una vez completado el mantenimiento del nodo, ejecute el comando **uncordons** para marcar el nodo como programable.

```
kubectl uncordon nodename
```

### ---Fin

### cluster-info

Para mostrar los complementos que se ejecutan en el clúster:

```
kubectl cluster-info
```

Para volcar la información actual del clúster a stdout:

```
kubectl cluster-info dump
```

### top\*

El comando **top** muestra el uso de recursos (CPU/memoria/almacenamiento). Este comando requiere que Heapster esté configurado correctamente y funcione en el servidor.

### taint\*

El comando **taint** actualiza las manchas en uno o más nodos.

### certificate\*

El comando **certificate** modifica los recursos del certificado.

## Comandos de diagnóstico y depuración de fallas

### describe

El comando **describe** es similar al comando **get**. La diferencia es que el comando **describe** muestra detalles de un recurso o grupo de recursos específicos, mientras que el comando **get** enumera uno o más recursos de un clúster. El comando **describe** no admite el indicador **-o**. Para los recursos del mismo tipo, los detalles de los recursos se imprimen en el mismo formato.

#### NOTA

Si se consulta la información sobre un recurso, puede utilizar el comando **get** para obtener información más detallada. Si desea comprobar el estado de un recurso específico, por ejemplo, para comprobar si un pod está en estado en ejecución, ejecute el comando **describe** para mostrar información de estado más detallada.

```
kubectl describe po <podname>
```

### logs

El comando **logs** imprime los logs de un contenedor en un pod o recurso especificado en stdout. Para mostrar los logs en el modo **tail -f**, ejecute este comando con el indicador **-f**.

```
kubectl logs -f podname
```

### exec

El comando de **kubectl exec** es similar al comando de Docker **exec** y ejecuta un comando en un contenedor. Si hay varios contenedores en un pod, utilice el indicador **-c** para elegir un pod.

```
kubectl exec -it podname bash  
kubectl exec -it podname -c containername bash
```

### port-forward\*

El comando **port-forward** reenvía uno o más puertos locales a un pod.

Por ejemplo:

Para escuchar en los puertos 5000 y 6000 localmente, reenviando datos hacia/desde los puertos 5000 y 6000 en el pod:

```
kubectl port-forward podname 5000:6000
```

### proxy\*

El comando **proxy** crea un servidor proxy entre localhost y el servidor de API de Kubernetes.

Por ejemplo:

Para habilitar las API de REST HTTP en el nodo principal:

```
kubectl proxy -accept-hosts= '*' -port=8001 -address= '0.0.0.0'
```

### cp

El comando **cp** copia archivos y directorios desde y hacia contenedores.

```
cp filename newfilename
```

### auth\*

El comando **auth** inspecciona la autorización.

**attach\***

El comando **attach** es similar al comando **logs -f** y se conecta a un proceso que ya se está ejecutando dentro de un contenedor existente. Para salir, ejecute el comando **ctrl-c**. Si un pod contiene los contenedores múltiples, para ver la salida de un contenedor específico, utilice el indicador **-c** y el *containername* después de *podname* para especificar un contenedor.

```
kubectl attach podname -c containername
```

## Comandos avanzados

**replace**

El comando **replace** actualiza o reemplaza un recurso existente por atributos, incluido el número de réplicas, etiquetas, versiones de imagen y puertos. Puede modificar directamente el archivo YAML original y luego ejecutar el comando **replace**.

```
kubectl replace -f filename
```

**AVISO**

Los nombres de recursos no se pueden actualizar.

**apply\***

El comando **apply** proporciona un control más estricto sobre la actualización de recursos que los comandos **patch** y **edit**. El comando **apply** aplica una configuración a un recurso y mantiene un conjunto de archivos de configuración en el control de origen. Siempre que hay una actualización, el archivo de configuración se envía al servidor y, a continuación, el comando **apply** kubectl aplica la última configuración al recurso. Kubernetes compara el nuevo archivo de configuración con el original y actualiza solo la configuración modificada en lugar de todo el archivo. La configuración que no está contenida en el indicador **-f** permanecerá sin cambios. A diferencia del comando **replace** que elimina el recurso y crea uno nuevo, el comando **apply** actualiza directamente el recurso original. Similar a la operación git, el comando **apply** agrega una anotación al recurso para marcar la aplicación actual.

```
kubectl apply -f
```

**patch**

Si desea modificar los atributos de un contenedor en ejecución sin eliminar primero el contenedor o usar el comando **replace**, el comando **patch** es para el rescate. El comando **patch** actualiza los campo(s) de un recurso mediante un parche de fusión estratégica, un parche de fusión JSON o un parche JSON. Por ejemplo, para cambiar una etiqueta de pod de **app=nginx1** a **app=nginx2** mientras el pod se está ejecutando, utilice el siguiente comando:

```
kubectl patch pod podname -p '{"metadata":{"labels":{"app":"nginx2}}}'
```

**convert\***

El comando **convert** convierte los archivos de configuración entre diferentes versiones de API.

## Comandos de configuración

**label**

El comando **label** actualiza las etiquetas de un recurso.

```
kubectl label pods my-pod new-label=newlabel
```

#### **annotate**

El comando **annotate** actualiza las anotaciones de un recurso.

```
kubectl annotate pods my-pod icon-url=http://.....
```

#### **completion**

El comando **completion** proporciona autocompletado para el shell.

## Otros Comandos

#### **api-versions**

El comando **api-versions** imprime las versiones de API admitidas.

```
kubectl api-versions
```

#### **api-resources**

El comando **api-resources** imprime los recursos de API admitidos.

```
kubectl api-resources
```

#### **config\***

El comando **config** modifica los archivos kubeconfig. Un ejemplo de caso de uso de este comando es configurar la información de autenticación en las invocaciones a la API.

#### **help**

El comando **help** obtiene todas las referencias de comandos.

#### **version**

El comando **version** imprime la información de la versión del cliente y del servidor para el contexto actual.

```
kubectl version
```

## 2.5 Actualización de un clúster

### 2.5.1 Descripción de la actualización

CCE ha aprobado el Certified Kubernetes Conformance Program y es una oferta certificada de Kubernetes. Para habilitar la interoperabilidad de una instalación de Kubernetes a la siguiente, debe actualizar sus clústeres de Kubernetes antes de que finalice el período de mantenimiento.

Después de que la última versión de Kubernetes esté disponible en CCE, CCE describirá los cambios en esta versión.

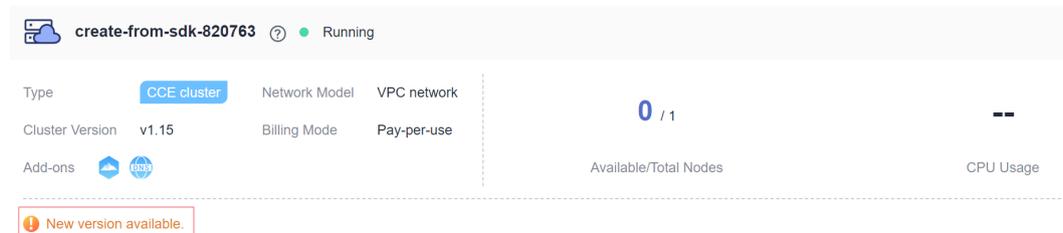
Puede utilizar la consola de CCE para actualizar la versión de Kubernetes de un clúster.

Se mostrará un indicador de actualización en la vista de tarjeta de clúster si hay una nueva versión para que el clúster actualice.

### Cómo comprobarlo:

Inicie sesión en la consola de CCE y compruebe si aparece el mensaje "New version available" en la esquina inferior izquierda del clúster. En caso afirmativo, se puede actualizar el clúster. Si no, el clúster no se puede actualizar.

**Figura 2-9** Clúster con el indicador de actualización



## Actualización de clúster

En la siguiente tabla se describe la versión de destino a la que se puede actualizar cada versión de clúster, los modos de actualización admitidos y los efectos de la actualización.

**Tabla 2-14** Rutas de actualización de clústeres e impactos

Versión de origen	Versión actualizada	Modos de actualización	Impactos
v1.19	v1.21	Actualización local	Necesita aprender acerca de las diferencias entre versiones. Para obtener más información, véase <a href="#">Precauciones para la actualización de la versión principal</a> .
v1.17 v1.15	v1.19	Actualización local	Necesita aprender acerca de las diferencias entre versiones. Para obtener más información, véase <a href="#">Precauciones para la actualización de la versión principal</a> .
v1.13	v1.15	Actualización gradual Actualización de reemplazar	<ul style="list-style-type: none"> <li>● <b>proxy</b> en el complemento de coredns no se puede configurar y necesita ser reemplazado por <b>forward</b>.</li> <li>● El complemento de almacenamiento se cambia de storage-driver a everest.</li> </ul>
v1.11 v1.9	La última versión que se puede crear en la consola	Migración	Necesita aprender acerca de las diferencias entre versiones. Para obtener más información, véase <a href="#">Precauciones para la actualización de la versión principal</a> .

Versión de origen	Versión actualizada	Modos de actualización	Impactos
v1.9.2 v1.7	La última versión que se puede crear en la consola	Migración	Necesita aprender acerca de las diferencias entre versiones. Para obtener más información, véase <a href="#">Precauciones para la actualización de la versión principal</a> .

## Modos de actualización

Los procesos de actualización son los mismos para los nodos maestros. Las diferencias entre los modos de actualización de los nodos de trabajo se describen de la siguiente manera:

**Tabla 2-15** Diferencias entre los modos de actualización y sus ventajas y desventajas

Modo de actualización	Método	Ventaja	Desventaja
<b>Actualización local</b>	Los componentes de Kubernetes, componentes de red y componentes de gestión de CCE se actualizan en el nodo. Durante la actualización, los pods de servicio y las redes no se ven afectados. La etiqueta <b>SchedulingDisabled</b> se agregará a todos los nodos existentes. Una vez completada la actualización, puede utilizar correctamente los nodos existentes.	No es necesario migrar servicios, lo que garantiza la continuidad del servicio.	La actualización in situ no actualiza el sistema operativo de un nodo. Si desea actualizar el sistema operativo, borre los datos de nodo correspondientes una vez completada la actualización del nodo y restablezca el nodo para actualizar el sistema operativo a una nueva versión.

Modo de actualización	Método	Ventaja	Desventaja
<p><b>Actualización gradual</b></p>	<p>Solo los componentes de Kubernetes y ciertos componentes de red se actualizan en el nodo. La etiqueta <b>SchedulingDisabled</b> se agregará a todos los nodos existentes para garantizar que las aplicaciones en ejecución no se vean afectadas.</p> <p><b>AVISO</b></p> <ul style="list-style-type: none"> <li>● <b>Una vez completada la actualización, debe crear manualmente nodos y liberar gradualmente los nodos antiguos</b> para migrar sus aplicaciones a los nuevos nodos. En este modo, puede controlar el proceso de actualización.</li> </ul>	<p>Los servicios no se interrumpen.</p>	<ul style="list-style-type: none"> <li>● <b>Una vez completada la actualización, debe crear manualmente nodos y liberar gradualmente los nodos antiguos.</b> Los nuevos nodos se facturan adicionalmente. Después de migrar los servicios a los nuevos nodos, se pueden eliminar los nodos antiguos.</li> <li>● Una vez completada la actualización continua, si desea continuar con la actualización a una versión posterior, primero debe restablecer los nodos antiguos. De lo contrario, no se puede pasar la comprobación previa a la actualización. Los servicios pueden ser interrumpidos durante la actualización.</li> </ul>
<p><b>Actualización de reemplazar</b></p>	<p>La última imagen del nodo de trabajo se utiliza para restablecer el sistema operativo del nodo.</p>	<p>Este es el modo de actualización más rápido y requiere pocas intervenciones manuales.</p>	<p>Los datos o configuraciones en el nodo se perderán, y los servicios se interrumpirán durante un período de tiempo.</p>

## Precauciones para la actualización de la versión principal

Ruta de actualización	Precaución	Autoverificación
v1.19 a v1.21	<p>El error de <b>exec probe timeouts</b> se corrige en Kubernetes 1.21. Antes de esta corrección de errores, la sonda exec no tiene en cuenta el campo <b>timeoutSeconds</b>. En su lugar, la sonda se ejecutará indefinidamente, incluso más allá de su fecha límite configurada. Se detendrá hasta que se devuelva el resultado. Si no se especifica este campo, se utiliza el valor predeterminado <b>1</b>. Este campo entra en vigor después de la actualización. Si el sondeo se ejecuta durante 1 segundo, la comprobación de estado de la aplicación puede fallar y la aplicación puede reiniciarse con frecuencia.</p>	<p>Antes de la actualización, compruebe si el tiempo de espera está configurado correctamente para la sonda exec.</p>
	<p>kube-apiserver de CCE 1.19 o posterior requiere que el campo Subject Alternative Names (SANs) esté configurado para el certificado de su servidor webhook. De lo contrario, kube-apiserver no puede invocar al servidor webhook después de la actualización, y contenedores no se puede iniciar correctamente.</p> <p>Causa raíz: X.509 <b>CommonName</b> se descarta en Go 1.15. kube-apiserver de CCE 1.19 se compila usando Go 1.15. Si su certificado webhook no tiene SAN, kube-apiserver no procesa el campo <b>CommonName</b> del certificado X.509 como nombre de host de forma predeterminada. Como resultado, la autenticación falla.</p>	<p>Antes de la actualización, compruebe si el campo SAN está configurado en el certificado de su servidor webhook.</p> <ul style="list-style-type: none"> <li>● Si no tiene su propio servidor webhook, puede omitir esta comprobación.</li> <li>● Si el campo no está definido, se recomienda utilizar el campo SAN para especificar la dirección IP y el nombre de dominio admitidos por el certificado.</li> </ul>

Ruta de actualización	Precaución	Autoverificación
<p>v1.15 a v1.19</p>	<p>El plano de control de en los clústeres v1.19 es incompatible con kubelet v1.15. Si un nodo no se puede actualizar o el nodo que se va a actualizar se reinicia después de que el nodo principal se actualice con éxito, hay una alta probabilidad de que el nodo esté en el estado <b>NotReady</b>.</p> <p>Esto se debe a que el nodo no puede actualizarse reinicia el kubelet y activa el registro del nodo. En los clústeres de v1.15, las etiquetas de registro predeterminadas <b>failure-domain.beta.kubernetes.io/is-baremetal</b> y <b>kubernetes.io/availablezone</b> son consideradas como etiquetas no válidas por los clústeres de v1.19.</p> <p>Las etiquetas válidas en los clústeres de v1.19 son <b>node.kubernetes.io/baremetal</b> y <b>failure-domain.beta.kubernetes.io/zone</b>.</p>	<ol style="list-style-type: none"> <li>1. En los casos normales, este escenario no se activa.</li> <li>2. Después de actualizar el nodo principal, no suspenda la actualización para que el nodo pueda actualizarse rápidamente.</li> <li>3. Si un nodo no se puede actualizar y no se puede restaurar, desaloje las aplicaciones en el nodo tan pronto como sea posible. Póngase en contacto con el soporte técnico y omita la actualización del nodo. Una vez completada la actualización, restablezca el nodo.</li> </ol>

Ruta de actualización	Precaución	Autoverificación
	<p>En los clústeres 1.15 y 1.19 de CCE, el sistema de archivos del controlador de almacenamiento de Docker cambia de XFS a Ext4. Como resultado, la secuencia de paquetes de importación en los pods de la aplicación Java actualizada puede ser anormal, causando excepciones de pod.</p>	<p>Antes de la actualización, compruebe el archivo de configuración de Docker <code>/etc/docker/daemon.json</code> en el nodo. Compruebe si el valor de <code>dm.fs</code> es de <code>xfs</code>.</p> <ul style="list-style-type: none"> <li>● Si el valor es de <code>ext4</code> o el controlador de almacenamiento es Overlay, puede omitir los siguientes pasos.</li> <li>● Si el valor es de <code>xfs</code>, se recomienda desplegar aplicaciones en el clúster de la nueva versión con antelación para probar si las aplicaciones son compatibles con la nueva versión del clúster.</li> </ul> <pre data-bbox="975 898 1430 1227"> {   "storage-driver":   "devicemapper",   "storage-opts": [     "dm.thinpooldev=/dev/mapper/     vgpaas-thinpool",     "dm.use_deferred_removal=true",     "dm.fs=xfs",     "dm.use_deferred_deletion=true"   ] }                     </pre>
	<p>kube-apiserver de CCE 1.19 o posterior requiere que el campo Subject Alternative Names (SANs) esté configurado para el certificado de su servidor webhook. De lo contrario, kube-apiserver no puede invocar al servidor webhook después de la actualización, y contenedores no se puede iniciar correctamente.</p> <p>Causa raíz: X.509 <code>CommonName</code> se descarta en Go 1.15. kube-apiserver de CCE 1.19 se compila usando Go 1.15. El campo <code>CommonName</code> se procesa como el nombre de host. Como resultado, la autenticación falla.</p>	<p>Antes de la actualización, compruebe si el campo SAN está configurado en el certificado de su servidor webhook.</p> <ul style="list-style-type: none"> <li>● Si no tiene su propio servidor webhook, puede omitir esta comprobación.</li> <li>● Si el campo no está definido, se recomienda utilizar el campo SAN para especificar la dirección IP y el nombre de dominio admitidos por el certificado.</li> </ul> <p><b>AVISO</b></p> <p>Para mitigar el impacto de las diferencias de versión en la actualización del clúster, CCE realiza un procesamiento especial durante la actualización de 1.15 a 1.19 y sigue soportando certificados sin SAN. Sin embargo, no se requiere ningún procesamiento especial para las actualizaciones posteriores. Le aconsejamos que rectifique su certificado lo antes posible.</p>

Ruta de actualización	Precaución	Autoverificación
	<p>En clústeres de v1.17.17 y posteriores, CCE crea automáticamente políticas de seguridad de pods (PSP) para usted, que restringen la creación de pods con configuraciones inseguras, por ejemplo, pods para los que <b>net.core.somaxconn</b> bajo un <code>sysctl</code> está configurado en el contexto de seguridad.</p>	<p>Después de una actualización, puede permitir configuraciones de sistema inseguras según sea necesario. Para obtener más información, véase <a href="#">Configuración de una política de seguridad de pod</a>.</p>
v1.13 a v1.15	<p>Después de actualizar un clúster de red de VPC, el nodo principal ocupa un bloque CIDR adicional debido a la actualización de los componentes de red. Si no hay ningún bloque CIDR contenedor disponible para el nuevo nodo, el pod programado para el nodo no puede ejecutarse.</p>	<p>Generalmente, este problema se produce cuando los nodos en el clúster están a punto de ocupar completamente el bloque CIDR contenedor. Por ejemplo, el bloque CIDR contenedor es 10.0.0.0/16, el número de direcciones IP disponibles es de 65,536 y a la red VPC se le asigna un bloque CIDR con el tamaño fijo (utilizando la máscara para determinar el número máximo de direcciones IP contenedor asignadas a cada nodo). Si el límite superior es 128, el clúster admite un máximo de 512 (65536/128) nodos, incluidos los tres nodos principales. Después de actualizar el clúster, cada uno de los tres nodos principales ocupa un bloque CIDR. Como resultado, se soportan 506 nodos.</p>

## 2.5.2 Antes de comenzar

Antes de la actualización, puede comprobar si el clúster se puede actualizar y qué versiones están disponibles en la consola de CCE. Para obtener más información, véase [Descripción de la actualización](#).

### Precauciones

- **Los clústeres actualizados no se pueden revertir. Por lo tanto, realice la actualización durante las horas no pico para minimizar el impacto en sus servicios.**
- Antes de actualizar un clúster, obtenga información sobre las características y diferencias de cada versión de clúster de [Notas del lanzamiento de Kubernetes](#) para evitar excepciones debido al uso de una versión de clúster incompatible.
- No **apagar, reiniciar o eliminar nodos** durante la actualización del clúster. De lo contrario, la actualización falla.

- Antes de actualizar un clúster, **deshabilite las políticas de ajuste automático** para evitar que se escala el nodo durante la actualización. De lo contrario, la actualización falla.
- Si modifica localmente la configuración de un nodo de clúster, la actualización del clúster puede fallar o la configuración puede perderse después de la actualización. Por lo tanto, modifique las configuraciones de la consola de CCE (página de lista de clústeres o grupos de nodos) para que se hereden automáticamente durante la actualización.
- Durante la actualización del clúster, los servicios de carga de trabajo en ejecución no se interrumpirán, pero el acceso al servidor de API se interrumpirá temporalmente.
- Antes de actualizar el clúster, compruebe si el clúster está en buen estado. Si el clúster es anormal, puede intentar rectificar el error. Si el error persiste, **envíe un ticket de servicio** para obtener ayuda.
- Para garantizar la seguridad de los datos, se recomienda realizar una copia de respaldo de los datos antes de actualizar el clúster. Durante la actualización, no se recomienda realizar ninguna operación en el clúster.
- Durante la actualización del clúster, la mancha **node.kubernetes.io/upgrade** (el efecto es de **NoSchedule**) se agrega al nodo. Una vez completada la actualización del clúster, se elimina la mancha. No agregue manchas con el mismo nombre de clave en el nodo. Incluso si las manchas tienen diferentes efectos, pueden ser eliminados por el sistema por error después de la actualización.

## Restricciones

- Actualmente, solo se pueden actualizar los clústeres de CCE que consisten en los nodos de VM y los clústeres de CCE Turbo. Los clústeres de Kunpeng solo se pueden actualizar a v1.19.
- Actualmente, los clústeres que utilizan imágenes privadas no se pueden actualizar.
- Después de actualizar el clúster, si la vulnerabilidad de containerd del motor de contenedor se corrige en las **Notas de la versión del clúster**, debe reiniciar manualmente containerd para que la actualización surta efecto. Lo mismo se aplica a los pods existentes.
- Si monta el archivo **docker.sock** en un nodo en un pod en modo HostPath, Docker se reiniciará durante la actualización, pero el archivo de calcetín en el contenedor no cambia. Como resultado, sus servicios pueden ser anormales, se recomienda montar el archivo de calcetín montando el directorio.
- Si se utiliza initContainer o Istio en la actualización in situ de un clúster de v1.15, preste atención a las siguientes restricciones:

En kubelet 1.16 y versiones posteriores, las **clases de QoS** son diferentes de las versiones anteriores. En kubelet 1.15 y versiones anteriores, solo se cuentan los contenedores en el **spec.containers**. En kubelet 1.16 y versiones posteriores, se cuentan los contenedores tanto en **spec.containers** como en **spec.initContainers**. La clase de QoS de un pod cambiará después de la actualización. Como resultado, el contenedor en el pod se reinicia. Se recomienda modificar la clase de QoS del contenedor de servicio antes de la actualización para evitar este problema. Para obtener más información, véase **Tabla 2-16**.

**Tabla 2-16** Cambios de clase de QoS antes y después de la actualización

Init Container (Calculado según spec.initContainers)	Service Container (Calculado según spec.containers)	Pod (Calculado según spec.containers y spec.initContainers)	Impactado o no
Guaranteed	Besteffort	Burstable	Sí
Guaranteed	Burstable	Burstable	No
Guaranteed	Guaranteed	Guaranteed	No
Besteffort	Besteffort	Besteffort	No
Besteffort	Burstable	Burstable	No
Besteffort	Guaranteed	Burstable	Sí
Burstable	Besteffort	Burstable	Sí
Burstable	Burstable	Burstable	No
Burstable	Guaranteed	Burstable	Sí

## Copia de respaldo de actualización

Cómo hacer una copia de respaldo de un nodo:

- Copia de respaldo de la base de datos etcd: CCE realiza automáticamente una copia de respaldo de la base de datos etcd durante la actualización del clúster.
- Copia de seguridad del nodo principal (recomendado **manual confirmation required**): En la página de confirmación de la actualización, haga clic en **Backup** para hacer una copia de respaldo de todo el nodo principal del clúster. El proceso de copia de respaldo utiliza el servicio Cloud Backup and Recovery (CBR) y tarda unos 20 minutos. Si hay muchas tareas de copia de respaldo en la nube en el sitio actual, el tiempo de copia de respaldo puede prolongarse.

## 2.5.3 Realización de la actualización in situ

### Escenario

Puede actualizar sus clústeres a una versión más reciente en la consola de CCE.

Antes de la actualización, obtenga información sobre la versión de destino a la que se puede actualizar cada clúster de CCE de qué manera y los impactos de la actualización. Para más detalles, véase [Descripción de la actualización](#) y [Antes de comenzar](#).

### Descripción

- Una actualización in situ actualiza los componentes de Kubernetes en los nodos del clúster, sin cambiar la versión de su sistema operativo.

- Los nodos del plano de datos se actualizan por lotes. De forma predeterminada, se priorizan en función de su CPU, memoria y **PodDisruptionBudgets (PDBs)**. También puede establecer las prioridades de acuerdo con sus requisitos de servicio.

## Precauciones

- Durante la actualización del clúster, el sistema actualizará automáticamente los complementos a una versión compatible con la versión del clúster de destino. No desinstale ni vuelva a instalar complementos durante la actualización del clúster.
- Antes de la actualización, asegúrese de que todos los complementos se estén ejecutando. Si un complemento no se actualiza, rectifique el error e inténtelo de nuevo.
- Durante la actualización, CCE comprueba el estado de ejecución del complemento. Algunos complementos (como coredns) requieren al menos dos nodos para ejecutarse normalmente. En este caso, al menos dos nodos deben estar disponibles para la actualización.
- Si se muestra un mensaje de error de actualización durante la actualización del clúster, rectifique el error como se le solicite e inténtelo de nuevo. Si los intentos de actualización fallan de nuevo, **envíe un ticket de servicio** para obtener ayuda.

Para obtener más información, consulte [Antes de comenzar](#).

## Procedimiento

La actualización del clúster pasa por la comprobación, la copia de respaldo, la configuración y la actualización y la verificación.

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** En el panel de navegación, elija **Cluster Upgrade**. Puede ver la versión recomendada a la derecha.

En la página **Cluster Upgrade**, compruebe la información de la versión, la hora de la última actualización/renovación, la versión de actualización disponible, el aviso y el historial de actualizaciones del clúster actual.

**Paso 3** Seleccione la versión del clúster que desea actualizar y haga clic en **Check**.

### NOTA

- Si el clúster tiene una nueva versión secundaria, no es necesario seleccionar la versión del clúster. La última versión secundaria se utiliza de forma predeterminada.
- Si el clúster tiene una nueva versión principal, puede seleccionar una versión según sea necesario.
- Si su clúster es de la última versión, la entrada de comprobación estará oculta.

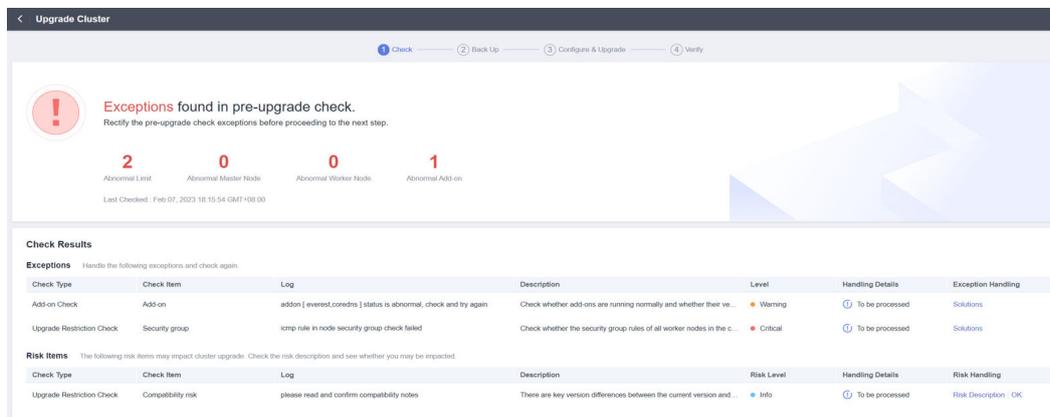


**Paso 4** Haga clic en **Start Check** y confirme la comprobación. Si hay elementos anormales o riesgosos en el clúster, maneje las excepciones en función de los resultados de comprobación que se muestran en la página y vuelva a comprobarlo.

- **Exceptions:** Vea la solución mostrada en la página, maneje las excepciones y vuelva a comprobar.

- **Risk Items:** puede afectar a la actualización del clúster. Verifique la descripción del riesgo y vea si puede verse afectado. Si no existe ningún riesgo, haga clic en **OK** junto al elemento de riesgo para omitir manualmente este elemento de riesgo y volver a comprobarlo.

Después de pasar la comprobación, haga clic en **Next: Back Up**.



**Paso 5** (Opcional) Copia de seguridad manual de los datos. Se realiza una copia de respaldo de los datos durante la actualización siguiendo una política predeterminada. Puede hacer clic en **Back Up** para realizar una copia de respaldo manual de los datos. Si no necesita realizar una copia de respaldo manual de los datos, haga clic en **Next: Configure & Upgrade**.

La copia de respaldo manual hará una copia de respaldo de todo el nodo principal. El proceso de copia de respaldo utiliza el servicio Cloud Backup and Recovery (CBR) y tarda unos 20 minutos. Si hay muchas tareas de copia de respaldo en la nube en el sitio actual, la copia de respaldo puede tardar más. No se puede actualizar el clúster durante la copia de respaldo.

**Paso 6** Configure los parámetros de actualización.

- **Add-on Upgrade Configuration:** Se enumeran los complementos que se han instalado en el clúster. Durante la actualización del clúster, el sistema actualiza automáticamente los complementos para que sean compatibles con la versión del clúster de destino. Puede hacer clic en **Set** para volver a definir los parámetros del complemento.

#### 📖 NOTA

Si se muestra un punto rojo ● a la derecha de un complemento, el complemento es incompatible con la versión del clúster de destino. Durante la actualización, el complemento se desinstalará y luego se volverá a instalar. Asegúrese de que los parámetros del complemento estén configurados correctamente.

- **Node Upgrade Configuration:** Puede establecer el número máximo de nodos que se van a actualizar en un lote.
- **Node Priority:** Puede establecer prioridades para los nodos que se van a actualizar. Si no establece este parámetro, el sistema determinará los nodos a actualizar por lotes según las condiciones específicas. Antes de establecer la prioridad de actualización de nodo, debe seleccionar un grupo de nodos. Los nodos y los grupos de nodos se actualizarán de acuerdo con las prioridades que especifique.
  - **Add Upgrade Priority:** Agregue prioridades de actualización para los grupos de nodos.
  - **Add Node Priority:** Después de agregar una prioridad de grupo de nodos, puede establecer la secuencia de actualización de nodos en el grupo de nodos. El sistema

actualiza los nodos en la secuencia que especifique. Si se omite esta configuración, el sistema actualiza los nodos según la política predeterminada.

**Paso 7** Una vez completada la configuración, haga clic en **Upgrade** y confirme la actualización. El clúster comienza a actualizarse. Puede ver el proceso en la parte inferior de la página.

Durante la actualización, puede hacer clic en **Suspend** a la derecha para suspender la actualización del clúster. Para continuar con la actualización, haga clic en **Continue**. Cuando la barra de progreso alcanza el 100%, se completa la actualización del clúster.

#### **NOTA**

Si se muestra un mensaje de error de actualización durante la actualización del clúster, rectifique el error como se le solicite e inténtelo de nuevo.

**Paso 8** Una vez completada la actualización, haga clic en **Next: Verify**. Verifique la actualización en función de los elementos de comprobación mostrados. Después de confirmar que todos los elementos de comprobación son normales, haga clic en **Complete** y confirme que se ha completado la comprobación posterior a la actualización.

Puede verificar la versión de Kubernetes del clúster en la página **Clusters**.

----Fin

## Preguntas frecuentes

- [¿Qué hago si un complemento de clúster no se actualiza durante la actualización del clúster de CCE?](#)

## 2.5.4 Actualización de sustitución/rodamiento (versión 1.13)

### Escenario

Puede actualizar sus clústeres a una versión más reciente de Kubernetes en la consola de CCE.

Antes de la actualización, obtenga información sobre la versión de destino a la que se puede actualizar cada clúster de CCE de qué manera y los impactos de la actualización. Para más detalles, véase [Descripción de la actualización](#) y [Antes de comenzar](#).

### Precauciones

- Si el complemento de coredns necesita actualizarse durante la actualización del clúster, asegúrese de que el número de nodos es mayor o igual que el número de instancias de coredns y todas las instancias de coredns se están ejecutando. De lo contrario, la actualización fallará. Antes de actualizar un clúster de v1.13, debe actualizar el complemento de coredns a la última versión disponible para el clúster.
- During the upgrade from one release of v1.13 to a later release of v1.13, applications in the cluster are interrupted for a short period of time only during the upgrade of network components.

### Procedimiento

**Paso 1** Inicie sesión en la consola de CCE y haga clic en el nombre del clúster para acceder al clúster.

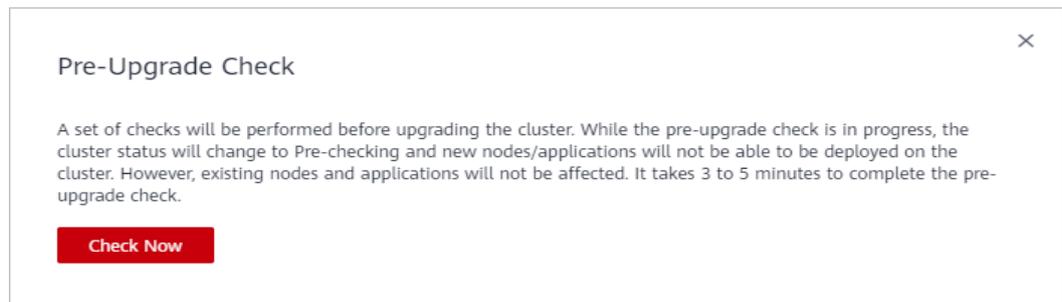
**Paso 2** En el panel de navegación, elija **Cluster Upgrade**. Puede ver la nueva versión disponible para la actualización a la derecha. Haga clic en **Upgrade**.

 **NOTA**

- Si la versión del clúster está actualizada, el botón **Upgrade** aparece atenuado.
- Si el estado del clúster es anormal o hay complementos anormales, el botón **Upgrade** está atenuado. Realice una comprobación haciendo referencia a [Antes de comenzar](#).

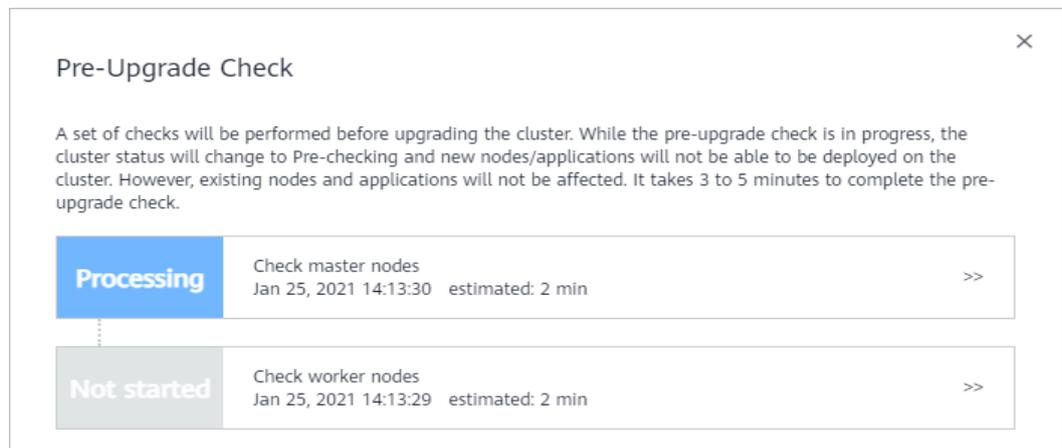
**Paso 3** En el cuadro de diálogo **Pre-upgrade Check** que se muestra, haga clic en **Check Now**.

**Figura 2-10** Comprobación previa a la actualización



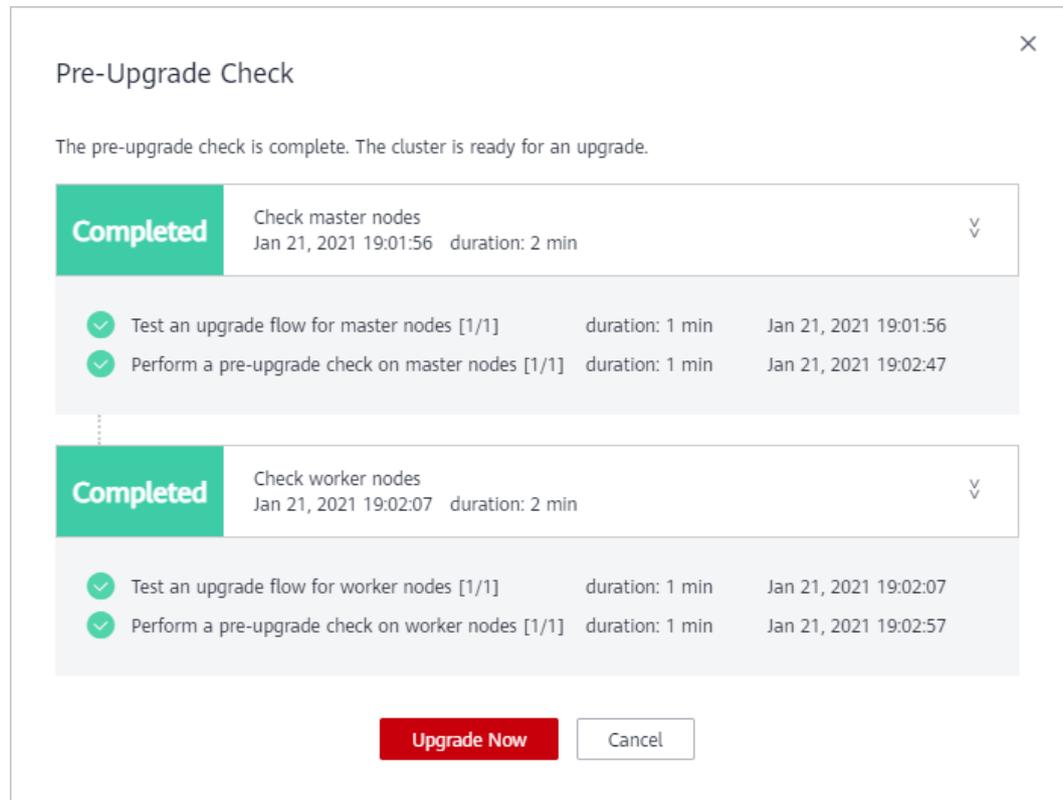
**Paso 4** Comienza la comprobación previa a la actualización. Mientras la comprobación previa a la actualización está en curso, el estado del clúster cambiará a **Pre-checking** y no se podrán implementar nuevos nodos/aplicaciones en el clúster. Sin embargo, los nodos y aplicaciones existentes no se verán afectados. Se tarda de 3 a 5 minutos en completar la comprobación previa a la actualización.

**Figura 2-11** Proceso de registro previo a la actualización



**Paso 5** Cuando el estado de la comprobación previa a la actualización es **Completed**, haga clic en **Upgrade**.

**Figura 2-12** Comprobación previa a la actualización completada



**Paso 6** En la página de actualización del clúster, revise o configure la información básica haciendo referencia a [Tabla 2-17](#).

**Tabla 2-17** Información básica

Parámetro	Descripción
Cluster Name	Revise el nombre del clúster que se va a actualizar.
Current Version	Revise la versión del clúster que se va a actualizar.
Target Version	Revise la versión de destino después de la actualización.

Parámetro	Descripción
Node Upgrade Policy	<p><b>Replace</b> (reemplazar la actualización): los nodos del trabajador se restablecerán. Sus sistemas operativos serán reinstalados, y los datos en el sistema y los discos de datos serán borrados. Tenga cuidado al realizar esta operación.</p> <p><b>NOTA</b></p> <ul style="list-style-type: none"> <li>● La función de gestión del ciclo de vida de los nodos y cargas de trabajo del clúster no está disponible.</li> <li>● Las API no se pueden invocar temporalmente.</li> <li>● Las cargas de trabajo en ejecución se interrumpirán porque los nodos se restablecen durante la actualización.</li> <li>● Los datos del sistema y los discos de datos de los nodos de trabajo se borrarán. Haga una copia de respaldo de los datos importantes antes de restablecer los nodos.</li> <li>● Los discos de datos sin LVM montados en los nodos de trabajo deben montarse de nuevo después de la actualización, y los datos de los discos no se perderán durante la actualización.</li> <li>● La cuota de disco de EVS debe ser mayor que 0.</li> <li>● Las direcciones IP de contenedor cambian, pero la comunicación entre contenedores no se ve afectada.</li> <li>● Se borrarán las etiquetas personalizadas de los nodos de trabajo.</li> <li>● Se tarda unos 12 minutos en completar la actualización del clúster.</li> </ul> <p><b>Gray</b> (actualización móvil): los nodos de trabajador se actualizan en modo continuo en un grupo de nodos. Este modo se aplica a escenarios en los que todos los nodos de un clúster se crean a partir de un grupo de nodos.</p> <p><b>NOTA</b></p> <ul style="list-style-type: none"> <li>● La función de gestión del ciclo de vida de los nodos y cargas de trabajo del clúster no está disponible.</li> <li>● Las API no se pueden invocar temporalmente.</li> <li>● Las cargas de trabajo en ejecución no se interrumpen.</li> <li>● Se tarda unos 12 minutos en completar la actualización del clúster.</li> </ul>
Login Mode	<p><b>Contraseña</b></p> <p>El nombre de usuario predeterminado es <b>root</b>. Introduzca la contraseña para iniciar sesión en el nodo y confirme la contraseña. Asegúrese de recordar la contraseña, ya que la necesitará cuando inicie sesión en el nodo.</p> <p><b>Par de claves</b></p> <p>Seleccione el par de claves utilizado para iniciar sesión en el nodo. Puede seleccionar una clave compartida.</p> <p>Se utiliza un par de claves para la autenticación de identidad cuando se inicia sesión de forma remota en un nodo. Si no hay ningún par de claves disponible, haga clic en <b>Create a Key Pair</b>. Para obtener más información sobre cómo crear un par de claves, consulte <a href="#">Creación de un par de claves</a>.</p>

Parámetro	Descripción
Cluster Backup	Se requiere una confirmación manual para realizar una copia de respaldo de todo el nodo principal. El proceso de copia de respaldo utiliza el servicio Cloud Backup and Recovery (CBR) y tarda unos 20 minutos. Si hay muchas tareas de copia de respaldo en la nube en el sitio actual, el tiempo de copia de respaldo puede prolongarse. Se recomienda hacer una copia de respaldo del nodo principal.
Node Upgrade Priority	Puede seleccionar los nodos que se van a actualizar primero.

**Paso 7** Haga clic en **Next**. En el cuadro de diálogo que aparece, haga clic en **OK**.

El mensaje que se muestra varía en función de la política de actualización de nodo seleccionada.

- **Replace**: Después de la actualización, el clúster utiliza sistemas operativos de una versión posterior. Durante la actualización, los nodos se reinician y los sistemas operativos se actualizan, lo que interrumpe los servicios.
- **Gray**: Necesita restablecer los nodos (y quite las etiquetas que hacen que los nodos sean impredecibles para los pods) o cree nodos para completar la actualización sucesiva.

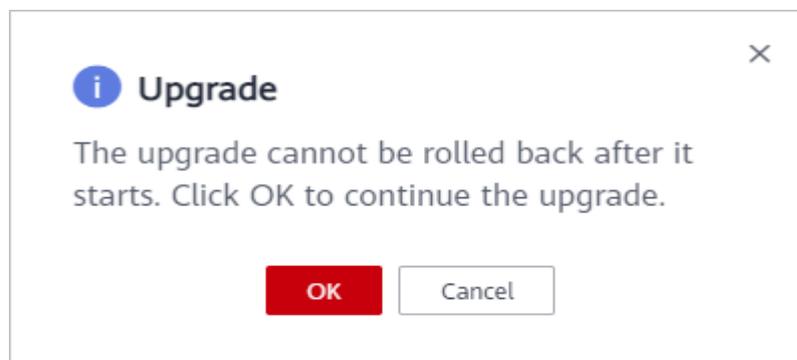
**Paso 8** Complementos de actualización. Si un complemento necesita ser actualizado, se muestra un punto rojo. Haga clic en el botón **Upgrade** en la esquina inferior izquierda de la vista de tarjeta adicional. Una vez completada la actualización, haga clic en **Upgrade** en la esquina inferior derecha de la página.

**NOTA**

- Los nodos maestros se actualizarán primero y, a continuación, los nodos de trabajo se actualizarán simultáneamente. Si hay un gran número de nodos de trabajo, se actualizarán en diferentes lotes.
- Seleccione una ventana de tiempo adecuada para la actualización para reducir el impacto en los servicios.
- Al hacer clic en **OK** se iniciará la actualización inmediatamente y la actualización no se puede cancelar. No apague ni reinicie los nodos durante la actualización.

**Paso 9** En el cuadro de diálogo **Upgrade** que se muestra, lea la información y haga clic en **OK**. Tenga en cuenta que el clúster no se puede revertir después de la actualización.

**Figura 2-13** Confirmación de la actualización del clúster



**Paso 10** De vuelta a la lista de clústeres, puede ver que el estado del clúster es **Upgrading**. Espere hasta que se complete la actualización.

Una vez que la actualización se realice correctamente, puede ver el estado y la versión del clúster en la lista de clústeres o en la página de detalles del clúster.

----Fin

## 2.5.5 Realización de la verificación posterior a la actualización

### 2.5.5.1 Verificación del servicio

#### Concepto de comprobación

Después de actualizar el clúster, compruebe si los servicios se están ejecutando de forma normal.

#### Procedimiento

Diferentes servicios tienen diferentes modos de verificación. Seleccione uno adecuado y verifique el servicio antes y después de la actualización.

Puede verificar el servicio desde los siguientes aspectos:

- La página de servicio está disponible.
- No se genera ninguna alarma o evento en la plataforma normal.
- No se genera ningún log de errores para los procesos clave.
- La prueba de marcación API es normal.

#### Solución

Si sus servicios en línea son anormales después de la actualización del clúster, póngase en contacto con el soporte técnico.

### 2.5.5.2 Comprobación de pod

#### Concepto de comprobación

- Compruebe si existen pods inesperados en el clúster.
- Compruebe si hay pods que se reinicien inesperadamente en el clúster.

#### Procedimiento

Vaya a la consola de CCE y acceda a la consola del clúster. Elija **Workloads** en el panel de navegación. En la página mostrada, cambie a la página de ficha **Pods**. Seleccione **All namespaces**, haga clic en **Status** y compruebe si existen pods anormales.



Pod Name	Status	Namespace	Pod IP	Node	Restarts	CPU Request/Limit/Usage	Memory Request/Limit/Usage	Created	Operation
condns-7f55c76c9c-bds78	Abnormal	kube-system	--	--	0	1 Cores 1 Cores 0.00%	1 GB 1 GB 0.00%	2 hours ago	Monitor   View Events   More
everest-csi-controller-6ddot	Abnormal	kube-system	--	--	0	0.25 Cores 0.25 Cores 0.00%	0.59 GB 1.46 GB 0.00%	2 hours ago	Monitor   View Events   More

Vea la columna **Restarts** para comprobar si hay pods que se reinicien de forma anormal.

Pod Name	Status	Namespace	Pod IP	Node	Restarts
coredns-78b88d65cc-bz9p	Abnormal	kube-system	--	--	0
coredns-78b88d65cc-pzq2j	Abnormal	kube-system	--	--	0
everest-csi-controller-7646	Abnormal	kube-system	--	--	0
everest-csi-controller-7646	Abnormal	kube-system	--	--	0

## Solución

Si hay pods anormales en el clúster después de la actualización del clúster, póngase en contacto con el soporte técnico.

### 2.5.5.3 Comprobación de red de nodos y contenedores

#### Concepto de comprobación

- Compruebe si los nodos se están ejecutando correctamente.
- Compruebe si la red del nodo es normal.
- Compruebe si la red de contenedor es normal.

#### Procedimiento

El estado del nodo refleja si el componente del nodo o la red es normal.

Vaya a la consola de CCE y acceda a la consola del clúster. Elija **Nodes** en el panel de navegación. Puede filtrar el estado de nodo por estado para comprobar si hay nodos anormales.

Node Name	Status	Node Pool	Configure
cce-test-nodepool-37811-rhdx	Running Schedulable	cce-test-nodepool-37811	AZ3 t6.xlarge.1 4vCPUs   4GiB
cce-test-nodepool-59938-k7es0	Running Non-schedulable	cce-test-nodepool-59938	AZ3 c7.large.2 2vCPUs   4GiB

La red de contenedor afecta a los servicios. Compruebe si sus servicios están disponibles.

## Solución

Si el estado del nodo es anormal, póngase en contacto con el soporte técnico.

Si la red de contenedor es anormal y sus servicios se ven afectados, póngase en contacto con el soporte técnico y confirme la ruta de acceso a la red anormal.

Origen	Destino	Tipo de destino	Posible falla
<ul style="list-style-type: none"> <li>● Pods (dentro de un clúster)</li> <li>● Nodos (dentro de un clúster)</li> <li>● Nodos en la misma VPC (fuera de un clúster)</li> <li>● Nubes de terceros</li> </ul>	Dirección IP pública del Service ELB	Entrada de balanceo de carga de tráfico de clúster	No hay registros.
	Dirección IP privada del Service ELB	Entrada de balanceo de carga de tráfico de clúster	No hay registros.
	Dirección IP pública de ingreso ELB	Entrada de balanceo de carga de tráfico de clúster	No hay registros.
	Dirección IP privada de ingreso ELB	Entrada de balanceo de carga de tráfico de clúster	No hay registros.
	Dirección IP pública del NodePort Service	Entrada de tráfico de clúster	Se sobrescribe la configuración del proxy kube. Este error se ha rectificado en el proceso de actualización.
	Dirección IP privada del NodePort Service	Entrada de tráfico de clúster	No hay registros.
	ClusterIP Service	Plano de red de Service	No hay registros.
	Puerto de Service no NodePort	Red de contenedores	No hay registros.
	Pods entre los nodos	Plano de red de contenedores	No hay registros.
	Pods en el mismo nodo	Plano de red de contenedores	No hay registros.
	Los nombres de dominio de Service y pod son resueltos por CoreDNS.	Resolución de nombres de dominio	No hay registros.

Origen	Destino	Tipo de destino	Posible falla
	Los nombres de dominio externos se resuelven según la configuración de hosts CoreDNS.	Resolución de nombres de dominio	Después de actualizar el complemento de coredns, se sobrescribe la configuración. Este error se ha rectificado en el proceso de actualización del complemento.
	Los nombres de dominio externos se resuelven según el servidor de flujo ascendente de CoreDNS.	Resolución de nombres de dominio	Después de actualizar el complemento de coredns, se sobrescribe la configuración. Este error se ha rectificado en el proceso de actualización del complemento.
	CoreDNS no resuelve los nombres de dominio externos.	Resolución de nombres de dominio	No hay registros.

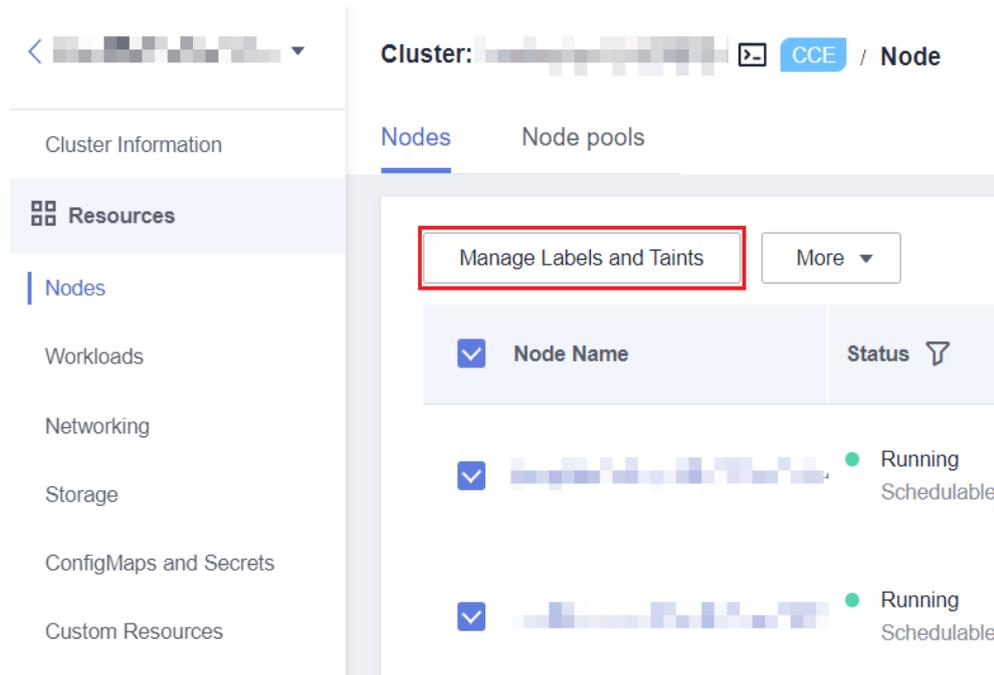
### 2.5.5.4 Comprobación de la etiqueta y la mancha del nodo

#### Concepto de comprobación

- Compruebe si se ha perdido la etiqueta.
- Compruebe si hay manchas inesperadas.

#### Procedimiento

Vaya a la consola de CCE, acceda a la consola del clúster y elija **Nodes** en el panel de navegación. En la página mostrada, haga clic en la ficha **Nodes**, seleccione todos los nodos y haga clic en **Manage Labels and Taints** para ver las etiquetas y manchas del nodo actual.



## Solución

Las etiquetas de usuario no se cambian durante la actualización del clúster. Si encuentra que las etiquetas se han perdido o se han agregado de forma anormal, póngase en contacto con el soporte técnico.

Si encuentra una mancha nueva (**node.kubernetes.io/upgrade**) en un nodo, es posible que el nodo se omita durante la actualización. Para obtener más información, véase [Comprobación de salto de nodo para restablecer](#).

Si encuentra que se agregan otras manchas al nodo, póngase en contacto con el soporte técnico.

### 2.5.5.5 Comprobación de nuevo nodo

#### Concepto de comprobación

Compruebe si se pueden crear nodos en el clúster.

#### Procedimiento

Vaya a la consola de CCE y acceda a la consola del clúster. Elija **Nodes** en el panel de navegación y haga clic en **Create Node**.



## Solución

Si no se pueden crear nodos en el clúster después de actualizarlo, póngase en contacto con el soporte técnico.

### 2.5.5.6 Comprobación de pod nuevo

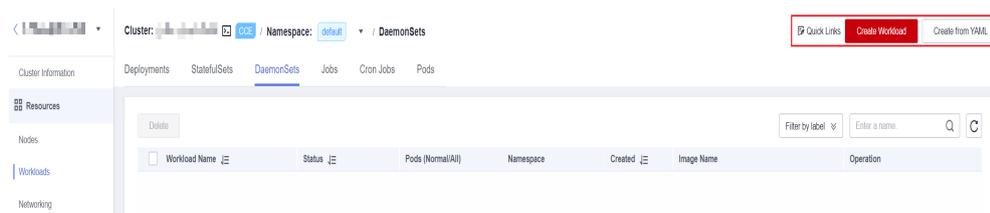
#### Concepto de comprobación

- Compruebe si se pueden crear pods en los nodos existentes después de actualizar el clúster.
- Compruebe si se pueden crear pods en nuevos nodos después de actualizar el clúster.

#### Procedimiento

Después de crear un nodo basado en [Comprobación de nuevo nodo](#), cree una carga de trabajo de DaemonSet para crear pods en cada nodo.

Vaya a la consola de CCE, acceda a la consola del clúster y elija **Workloads** en el panel de navegación. En la página mostrada, cambie a la pestaña **DaemonSets** y haga clic en **Create Workload** o **Create from YAML** en la esquina superior derecha.



Se recomienda utilizar la imagen para las pruebas de rutina como la imagen base. Puede desplegar un pod haciendo referencia al siguiente archivo YAML.

#### NOTA

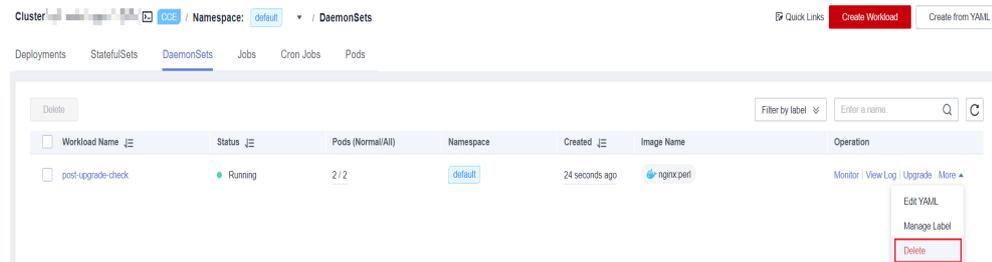
En esta prueba, YAML despliega DaemonSet en el espacio de nombres predeterminado, utiliza **nginx:perl** como imagen base, solicita 10 MB de CPU y 10 Mi de memoria, y limita 100 MB de CPU y 50 Mi de memoria.

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: post-upgrade-check
  namespace: default
spec:
  selector:
    matchLabels:
      app: post-upgrade-check
      version: v1
  template:
    metadata:
      labels:
        app: post-upgrade-check
        version: v1
    spec:
      containers:
        - name: container-1
          image: nginx:perl
          imagePullPolicy: IfNotPresent
          resources:
            requests:
```

```
cpu: 10m
memory: 10Mi
limits:
  cpu: 100m
  memory: 50Mi
```

Una vez creada la carga de trabajo, compruebe si el estado del pod de la carga de trabajo es normal.

Una vez completada la comprobación, vaya a la consola de CCE y acceda a la consola del clúster. Elija **Workloads** en el panel de navegación. En la página mostrada, cambie a la página de ficha **DaemonSets**, elija **More > Delete** en la columna **Operation** de la carga de trabajo de **post-upgrade-check** para eliminar la carga de trabajo de prueba.



## Solución

Si el pod no se puede crear o el estado del pod es anormal, póngase en contacto con el soporte técnico y especifique si la excepción se produce en nodos nuevos o existentes.

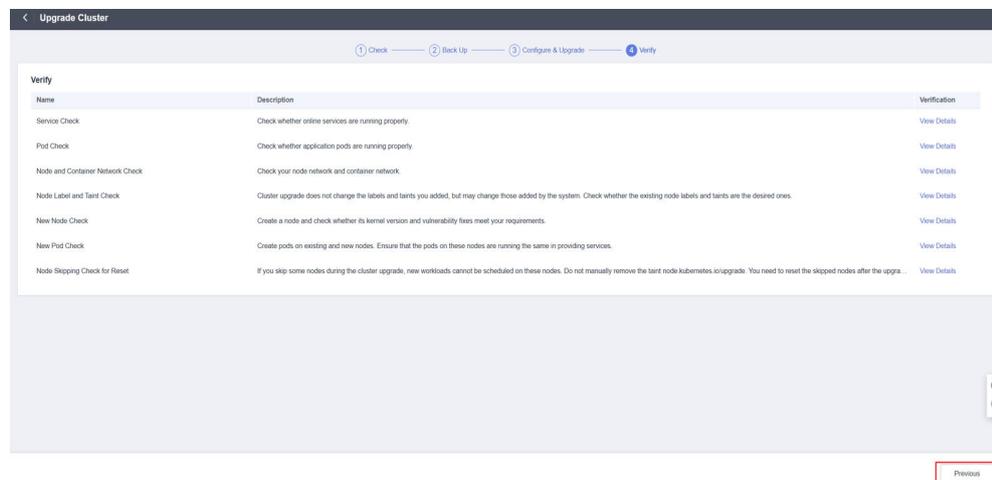
### 2.5.5.7 Comprobación de salto de nodo para restablecer

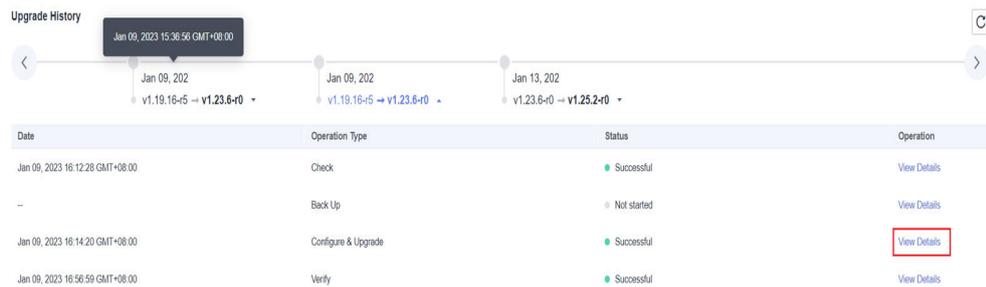
#### Concepto de comprobación

Después de actualizar el clúster, debe restablecer los nodos que no se pueden actualizar.

#### Procedimiento

Vuelva al paso anterior o vea los detalles de la actualización en la página de historial de actualizaciones para ver los nodos que se omiten durante la actualización.





Los nodos omitidos se muestran en la página de detalles de actualización. Restablezca los nodos omitidos una vez completada la actualización. Para obtener más información acerca de cómo restablecer un nodo, consulte [Restablecimiento de un nodo](#).

**NOTA**

Al restablecer un nodo se restablecerán todas las etiquetas de nodo, lo que puede afectar a la programación de la carga de trabajo. Antes de restablecer un nodo, compruebe y conserve las etiquetas que haya agregado manualmente al nodo.

## 2.5.6 Migración de servicios a través de clústeres de diferentes versiones

### Escenarios de aplicación

Esta sección describe cómo migrar servicios de un clúster de una versión anterior a un clúster de una versión posterior en CCE.

Esta operación es aplicable cuando se requiere una actualización de clúster entre versiones (por ejemplo, la actualización de v1.7.\* o v1.9.\* a 1.17.\*) y se pueden crear nuevos clústeres para la migración de servicios.

### Requisitos previos

**Tabla 2-18** Lista de comprobación antes de la migración

Categoría	Descripción
Cluster	Relacionado con NodeIP: Compruebe si las direcciones IP de nodo (incluidos las EIP) del clúster antes de la migración se han utilizado en otras configuraciones o listas blancas.
Workloads	Registre el número de cargas de trabajo para la comprobación posterior a la migración.
Storage	<ol style="list-style-type: none"> <li>Compruebe si los recursos de almacenamiento en uso son aprovisionados por la nube o por su organización.</li> <li>Cambie el almacenamiento creado automáticamente al almacenamiento existente en el nuevo clúster.</li> </ol>

Categoría	Descripción
Network	<ol style="list-style-type: none"> <li>1. Preste especial atención al ELB y al ingreso.</li> <li>2. Los clústeres de una versión anterior solo admiten el balanceador de carga clásico. Para migrar servicios a un nuevo clúster, debe cambiar el tipo de balanceador de carga a balanceador de carga compartido. A continuación, se restablecerá el servicio de ELB correspondiente.</li> </ol>
O&M	Configuración privada: Compruebe si los parámetros del núcleo o los datos del sistema se han configurado en los nodos del clúster.

## Procedimiento

### Paso 1 Crear un clúster de CCE.

Cree un clúster con las mismas especificaciones y configuraciones que el clúster de la versión anterior. Para obtener más información, véase [Compra de un clúster de CCE](#).

### Paso 2 Agregar un nodo.

Agregue nodos con las mismas especificaciones y elementos de configuración manual. Para obtener más información, véase [Creación de un nodo](#).

### Paso 3 Crear un volumen de almacenamiento en el nuevo clúster.

Utilice un volumen de almacenamiento existente para crear un PVC en el nuevo clúster. El nombre de PVC no cambia. Para más detalles, consulte [Uso de un bucket de OBS existente con un PV estático](#) o [Uso de un sistema de archivos de SFS Turbo existente con un PV estático](#).

#### NOTA

La conmutación de almacenamiento solo admite bucket de OBS y sistemas de archivos de SFS Turbo. Si se utiliza una conmutación de almacenamiento no compartida, debe suspender las cargas de trabajo del clúster antiguo para cambiar los recursos de almacenamiento. Como resultado, los servicios no estarán disponibles.

### Paso 4 Crear una carga de trabajo en el nuevo clúster.

El nombre y las especificaciones de la carga de trabajo permanecen sin cambios. Para obtener más información sobre cómo crear una carga de trabajo, consulte [Creación de una Deployment](#) o [Creación de un StatefulSet](#).

### Paso 5 Volver a montar el almacén.

Vuelva a montar el almacenamiento existente en la carga de trabajo. Para más detalles, consulte [Uso de un bucket de OBS existente con un PV estático](#) o [Uso de un sistema de archivos de SFS Turbo existente con un PV estático](#).

### Paso 6 Crear un Service en el nuevo clúster.

El nombre del Service y las especificaciones permanecen sin cambios. Para obtener más información acerca de cómo crear un Service, consulte [Service](#).

### Paso 7 Servicios de la comisión.

Una vez que se hayan creado todos los recursos, comisione los servicios en contenedores. Si la puesta en marcha se realiza correctamente, migre los servicios al nuevo clúster.

**Paso 8 Eliminar or darse de baja del antiguo clúster.**

Cuando todas las funciones del nuevo clúster sean estables, cancele la suscripción o elimine el clúster antiguo. Para obtener más información sobre cómo eliminar un clúster, consulte [Eliminación de un clúster](#).

---Fin

## 2.5.7 Solución de problemas de excepciones de comprobación previa a la actualización

### 2.5.7.1 Realización de la comprobación previa a la actualización

El sistema realiza una completa comprobación previa a la actualización antes de la actualización del clúster. Si el clúster no cumple las condiciones de comprobación previa a la actualización, la actualización no puede continuar. Para evitar riesgos de actualización, puede realizar una comprobación previa a la actualización de acuerdo con los elementos de comprobación proporcionados en esta sección.

**Tabla 2-19** Conceptos de comprobación

Concepto de comprobación	Descripción
<b>Comprobación del nodo</b>	<ul style="list-style-type: none"> <li>● Compruebe si el nodo está disponible.</li> <li>● Compruebe si el sistema operativo del nodo admite la actualización.</li> <li>● Compruebe si hay etiquetas de grupo de nodos inesperadas en el nodo.</li> <li>● Compruebe si el nombre del nodo de Kubernetes es coherente con el nombre de ECS.</li> </ul>
<b>Comprobación de la lista de bloqueo</b>	Compruebe si el usuario actual está en la lista de bloqueo de actualización.
<b>Comprobación del complemento</b>	<ul style="list-style-type: none"> <li>● Compruebe si el estado del complemento es normal.</li> <li>● Compruebe si el complemento admite la versión de destino.</li> </ul>
<b>Comprobación del gráfico de Helm</b>	Compruebe si el registro de HelmRelease actual contiene las API de Kubernetes descartadas que no son compatibles con la versión del clúster de destino. En caso afirmativo, es posible que el gráfico de Helm no esté disponible después de la actualización.
<b>Comprobación de la conectividad SSH del nodo principal</b>	Compruebe si CCE puede conectarse a sus nodos principales.

Concepto de comprobación	Descripción
<b>Comprobación del grupo de nodos</b>	<ul style="list-style-type: none"> <li>● Compruebe el estado del nodo.</li> <li>● Compruebe si la función de ajuste automático del grupo de nodos está deshabilitada.</li> </ul>
<b>Comprobación del grupo de seguridad</b>	Compruebe si el grupo de seguridad permite que el nodo principal acceda a los nodos mediante ICMP.
<b>Restricción del nodo de Arm</b>	<ul style="list-style-type: none"> <li>● Compruebe si el clúster es un clúster de Kunpeng o si el clúster híbrido contiene los nodos principales basados en Arm.</li> <li>● Compruebe si el clúster contiene los nodos de Arm.</li> </ul>
<b>Nodo por migrar</b>	Compruebe si el nodo necesita ser migrado.
<b>Recurso de Kubernetes descartado</b>	Compruebe si hay recursos descartados en los clústeres.
<b>Riesgo de compatibilidad</b>	<p>Compruebe si la versión de Kubernetes tiene diferencias de compatibilidad.</p> <p>La actualización del parche no implica diferencias de compatibilidad de versiones.</p>
<b>Versión de nodo de CCEAgent</b>	Compruebe si cce-agent en el nodo actual es de la versión más reciente.
<b>Uso de la CPU del nodo</b>	Compruebe si el uso de CPU del nodo excede el 90%.
<b>Comprobación de CRD</b>	<ul style="list-style-type: none"> <li>● Compruebe si se ha eliminado <b>packageversions.version.cce.io</b> de CRD clave del clúster.</li> <li>● Compruebe si se ha eliminado <b>network-attachment-definitions.k8s.cni.cncf.io</b> de CRD de clave de clúster.</li> </ul>
<b>Disco de nodo</b>	<ul style="list-style-type: none"> <li>● Compruebe si el uso de discos de datos clave en el nodo cumple con los requisitos de actualización.</li> <li>● Compruebe si el directorio <b>/tmp</b> tiene 500 MB de espacio disponible.</li> </ul>
<b>Nodo de DNS</b>	<ul style="list-style-type: none"> <li>● Compruebe si la configuración de DNS del nodo actual puede resolver la dirección de OBS.</li> <li>● Compruebe si el nodo actual puede acceder a la dirección de OBS del paquete de componentes de actualización de almacenamiento.</li> </ul>
<b>Permisos de archivo de directorio de clave de nodo</b>	Compruebe si el directorio de claves <b>/var/paas</b> en los nodos contiene archivos con propietarios o grupos de propietarios anormales.
<b>Kubelet</b>	Compruebe si el kubelet del nodo se está ejecutando correctamente.

Concepto de comprobación	Descripción
<b>Memoria de nodos</b>	Compruebe si el uso de memoria del nodo supera el 90%.
<b>Servidor de sincronización de reloj de nodo</b>	Compruebe si el servidor de sincronización de reloj ntpd o chronyd del nodo se está ejecutando correctamente.
<b>SO del nodo</b>	Compruebe si la versión del kernel del sistema operativo del nodo es compatible con CCE.
<b>Recuento de CPU de nodo</b>	Compruebe si el número de CPUs en el nodo principal es mayor que 2.
<b>Comando de nodo de Python</b>	Compruebe si los comandos de Python están disponibles en un nodo.
<b>Versión de ASM</b>	<ul style="list-style-type: none"> <li>● Compruebe si el clúster utiliza ASM.</li> <li>● Compruebe si la versión actual de ASM admite la versión del clúster de destino.</li> </ul>
<b>Preparación del nodo</b>	Compruebe si los nodos del clúster están listos.
<b>Diario de nodo</b>	Compruebe si el diario de un nodo es normal.
<b>containerd.sock</b>	Compruebe si el archivo <b>containerd.sock</b> existe en el nodo. Este archivo afecta al inicio del tiempo de ejecución de contenedor en el Euler OS.
<b>Error interno</b>	Antes de la actualización, compruebe si se produce un error interno.
<b>Punto de montaje del nodo</b>	Compruebe si existen puntos de montaje inaccesibles en el nodo.
<b>Mancha de nodo de Kubernetes</b>	Compruebe si existen manchas necesarias para la actualización del clúster en el nodo.
<b>Restricción de everest</b>	Compruebe si el complemento de everest actual tiene las restricciones de compatibilidad.
<b>Restricción de cce-hpa-controller</b>	Compruebe si el complemento de cce-controller-hpa actual tiene restricciones de compatibilidad.
<b>Enlace mejorado del núcleo de la CPU</b>	Compruebe si la versión actual del clúster y la versión de destino admiten un enlace mejorado del núcleo de la CPU.
<b>Estado de componentes de nodo de usuario</b>	Compruebe si el tiempo de ejecución contenedor y los componentes de red en el nodo de usuario están en buen estado.
<b>Estado de los componentes del nodo del controlador</b>	Compruebe si los componentes de Kubernetes, tiempo de ejecución de contenedor y red del nodo del controlador están en buen estado.

Concepto de comprobación	Descripción
<b>Límite de recursos de memoria del componente de Kubernetes</b>	Compruebe si los recursos de los componentes de Kubernetes, como etcd y kube-controller-manager, exceden el límite superior.
<b>API de Kubernetes descartadas</b>	Compruebe si la API invocada ha sido descartada.
<b>Capacidad de IPv6 de un clúster de CCE Turbo</b>	Si IPv6 está habilitado para un clúster de CCE Turbo, compruebe si la versión del clúster de destino admite IPv6.
<b>NetworkManager de nodo</b>	Compruebe el estado del NetworkManager.
<b>Archivo de ID de nodo</b>	Compruebe el formato de archivo ID.
<b>Consistencia de la configuración del nodo</b>	Cuando actualice un clúster de CCE a v1.19 o posterior, se comprobará la configuración del componente de Kubernetes en su nodo.
<b>Archivo de configuración de nodo</b>	Compruebe si los archivos de configuración de los componentes clave existen en el nodo.
<b>Consistencia de la configuración de CoreDNS</b>	Compruebe si la configuración actual de la clave de CoreDNS Corefile es diferente del registro de lanzamiento de Helm. La diferencia puede sobrescribirse durante la actualización del complemento, <b>afecta la resolución de nombres de dominio en el clúster.</b>

## 2.5.7.2 Comprobación del nodo

### Concepto de comprobación

Compruebe los siguientes aspectos:

- Compruebe si el nodo está disponible.
- Compruebe si el sistema operativo del nodo admite la actualización.
- Compruebe si hay etiquetas de grupo de nodos inesperadas en el nodo.
- Compruebe si el nombre del nodo de Kubernetes es coherente con el nombre de ECS.

### Solución

- **Escenario 1: El nodo no es disponible.**

Inicie sesión en la consola de CCE y acceda a la consola del clúster. Elija **Nodos** en el panel de navegación y compruebe el estado del nodo. Asegúrese de que el nodo está en el estado **Running**. No se puede actualizar un nodo con el estado **Installing** o **Deleting**.

Si el estado del nodo es anormal, restaure el nodo haciendo referencia a **¿Qué debo hacer si un clúster está disponible pero algunos nodos no están disponibles?** y vuelva a intentar la tarea de comprobación.



● **Escenario 2: El SO del nodo no admite la actualización.**

En la siguiente tabla se enumeran los sistemas operativos de nodo que admiten la actualización. Puede restablecer el sistema operativo del nodo a un sistema operativo disponible en la lista.

**Tabla 2-20** Sistemas operativos que admiten la actualización

SO	Restricción
EulerOS 2.3/2.5/2.8/2.9/2.10	Ninguna.
CentOS 7.6/7.7	Ninguna.
Ubuntu 18.04/22.04	Algunos sitios no pueden realizar actualizaciones. Si el resultado de la comprobación muestra que la actualización no es compatible, póngase en contacto con el soporte técnico.
Huawei Cloud EulerOS 2.0	Algunos sitios no pueden realizar actualizaciones. Si el resultado de la comprobación muestra que la actualización no es compatible, póngase en contacto con el soporte técnico.

● **Escenario 3: Hay etiquetas de grupo de nodos inesperadas en el nodo.**

Si se migra un nodo desde un grupo de nodos al grupo de nodos predeterminado, se conserva la etiqueta de grupo de nodos **cce.cloud.com/cce-nodepool**, lo que afecta a la actualización del clúster. Compruebe si la programación de carga en el nodo depende de la etiqueta.

- Si no hay dependencia, elimine la etiqueta.
- En caso afirmativo, modifique la política de equilibrio de carga, quite la dependencia y, a continuación, elimine la etiqueta.

### 2.5.7.3 Comprobación de la lista de bloqueo

#### Concepto de comprobación

Compruebe si el usuario actual está en la lista de bloqueo de actualización.

## Solución

CCE deshabilita temporalmente la función de actualización del clúster debido a las siguientes razones:

- El clúster se identifica como el clúster principal de producción.
- Se están realizando o se realizarán otras tareas de O&M para mejorar la estabilidad del clúster, por ejemplo, la reconstrucción 3AZ del nodo principal.

Puede ponerse en contacto con el soporte técnico.

### 2.5.7.4 Comprobación del complemento

#### Concepto de comprobación

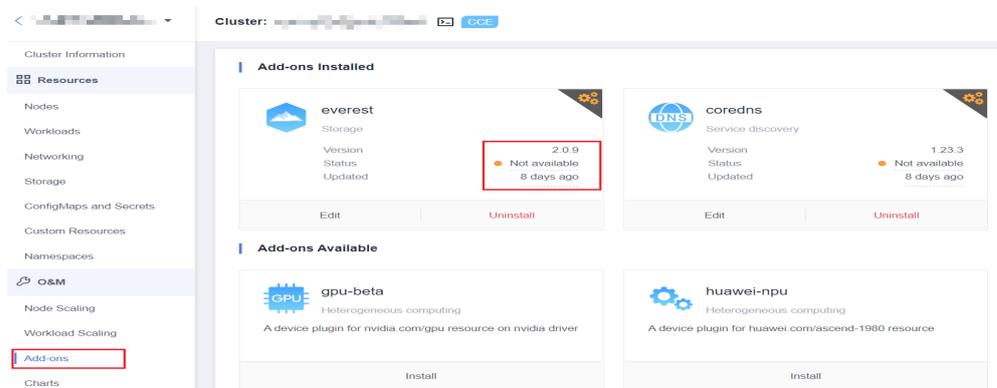
Compruebe los siguientes aspectos:

- Compruebe si el estado del complemento es normal.
- Compruebe si el complemento admite la versión de destino.

## Solución

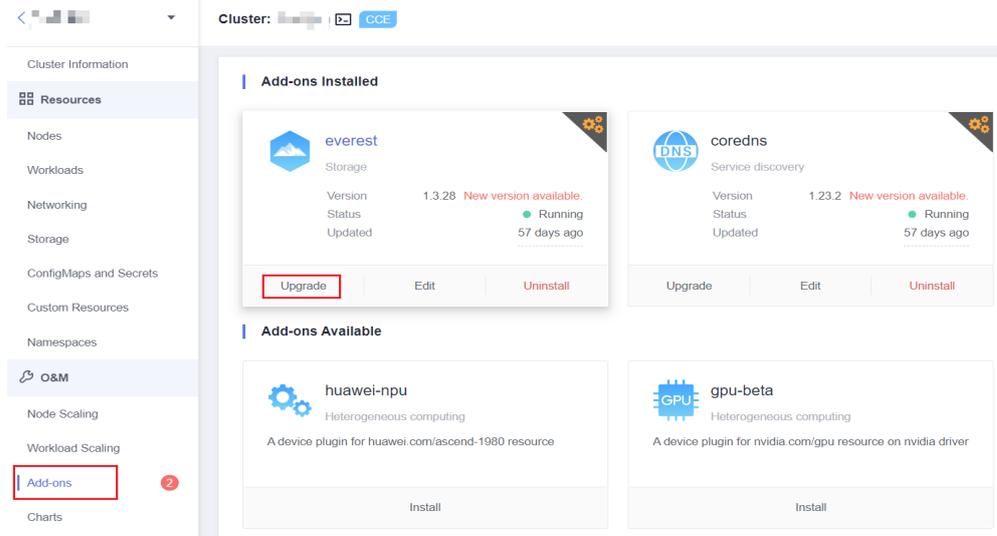
- **Escenario 1: El estado del complemento es anormal.**

Inicie sesión en la consola de CCE y vaya al clúster de destino. Elija **O&M > Add-ons** para ver y manejar el complemento anormal.



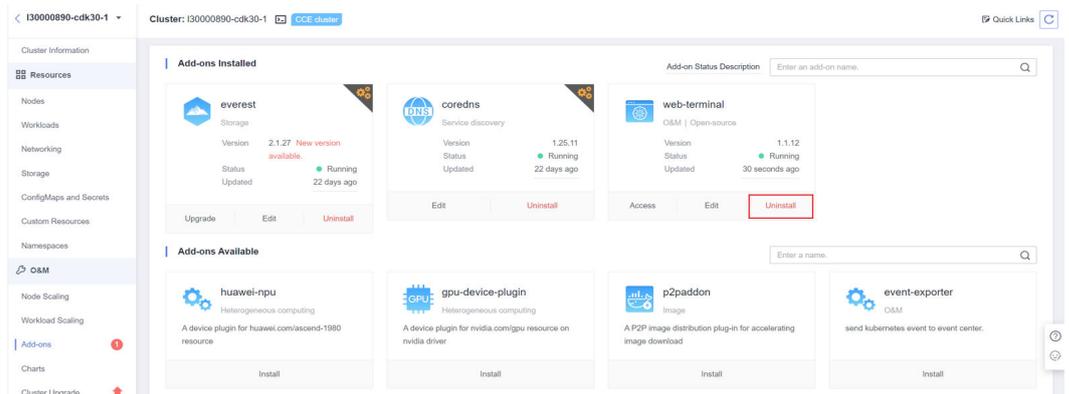
- **Escenario 2: La versión de destino no admite el complemento actual.**

El complemento no se puede actualizar automáticamente con el clúster. Inicie sesión en la consola de CCE y vaya al clúster de destino. Elija **O&M > Add-ons** para actualizar manualmente el complemento.



- **Escenario 3: Después de actualizar el complemento a la versión más reciente, la versión del clúster de destino aún no se admite.**

Inicie sesión en la consola de CCE y vaya al clúster de destino. Elija **O&M > Add-ons** para desinstalar manualmente el complemento. Para obtener más información sobre las versiones de complementos y las soluciones de reemplazo compatibles, consulte el documento de [Ayuda](#).



### 2.5.7.5 Comprobación del gráfico de Helm

#### Concepto de comprobación

Compruebe si el registro de HelmRelease actual contiene las API de Kubernetes descartadas que no son compatibles con la versión del clúster de destino. En caso afirmativo, es posible que el gráfico de Helm no esté disponible después de la actualización.

#### Solución

Convierta las API de Kubernetes descartadas en las API compatibles con las versiones de origen y destino.

#### 📖 NOTA

Este artículo se ha procesado automáticamente en el proceso de actualización. Puede ignorar este elemento.

## 2.5.7.6 Comprobación de la conectividad SSH del nodo principal

### Concepto de comprobación

Compruebe si CCE puede conectarse a sus nodos principales.

### Solución

Contacte con el servicio de asistencia técnica.

## 2.5.7.7 Comprobación del grupo de nodos

### Concepto de comprobación

Compruebe los siguientes aspectos:

- Compruebe el estado del nodo.
- Compruebe si la función de ajuste automático del grupo de nodos está deshabilitada.

### Solución

- **Escenario 1: El estado del grupo de nodo es anormal.**

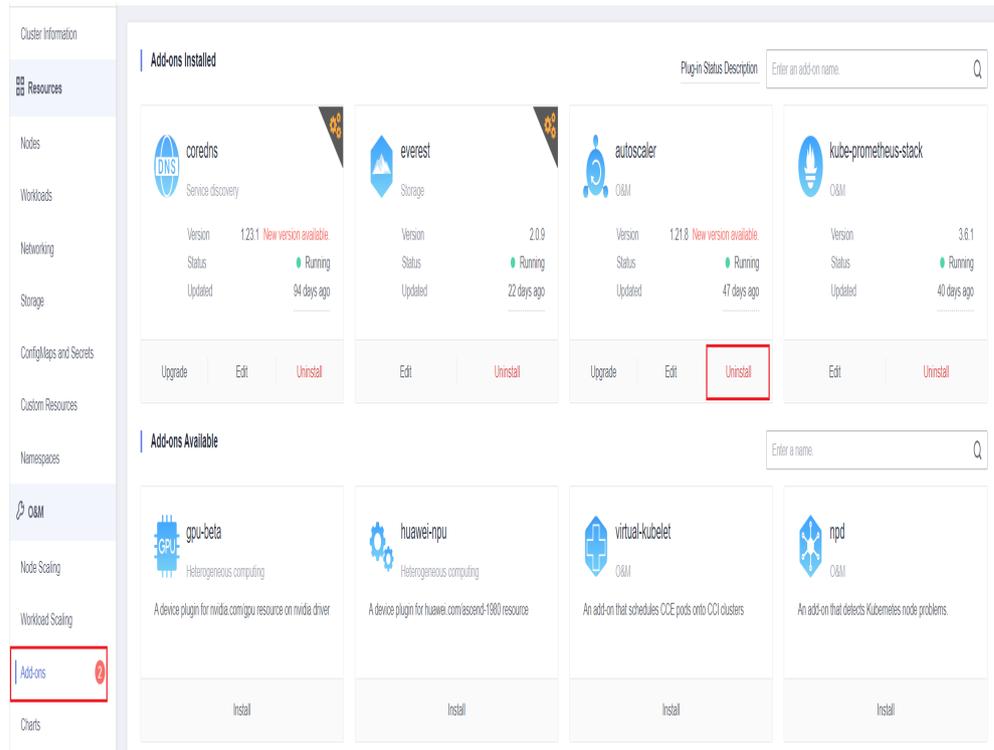
Inicie sesión en la consola de CCE, vaya al clúster de destino y elija **Nodos**. En la página mostrada, haga clic en la ficha **Node Pools** y compruebe el estado del grupo de nodos. Si se está escalando el grupo de nodos, espere hasta que se complete el ajuste y desactive la función de ajuste automático haciendo referencia al **Escenario 2**.



- **Escenario 2: Se activa la función del ajuste automático del grupo de nodo.**

#### Solución 1 (Recomendada)

Inicie sesión en la consola de CCE y vaya al clúster de destino. Elija **O&M > Add-ons** y desinstale el complemento del escalador automático.



### NOTA

Antes de desinstalar el complemento del escalador automático, haga clic en **Upgrade** para hacer una copia de respaldo de la configuración para que la configuración del complemento se pueda restaurar durante la reinstalación.

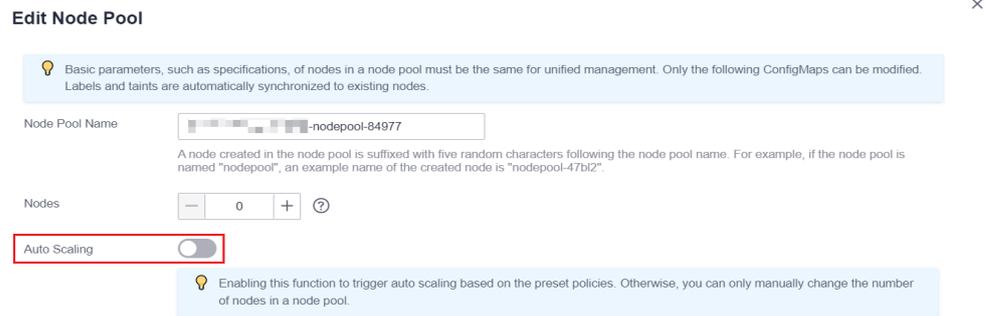
Antes de desinstalar el complemento del escalador automático, elija **O&M > Node Scaling** y realice una copia de respaldo de las políticas de ajuste actuales para que puedan restaurarse durante la reinstalación. Estas políticas se eliminarán cuando se desinstala el complemento del escalador automático.

Para obtener y realizar una copia de respaldo de la política de ajuste de nodos, haga clic en **Edit**.



### Solución 2

Si no desea desinstalar el complemento del escalador automático, inicie sesión en la consola de CCE y acceda a la página de detalles del clúster. Elija **Nodes** en el panel de navegación. En la página mostrada, haga clic en la ficha **Node Pools** y haga clic en **Edit** del grupo de nodos correspondiente para desactivar la función de ajuste automático.



**NOTA**

Antes de desactivar la función de ajuste automático, haga una copia de respaldo de la configuración de ajuste automático para que la configuración pueda restaurarse cuando la función esté activada.

### 2.5.7.8 Comprobación del grupo de seguridad

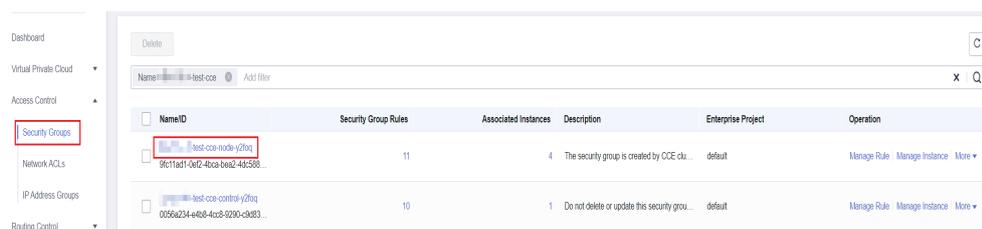
#### Concepto de comprobación

Compruebe si el grupo de seguridad permite que el nodo principal acceda a los nodos mediante ICMP.

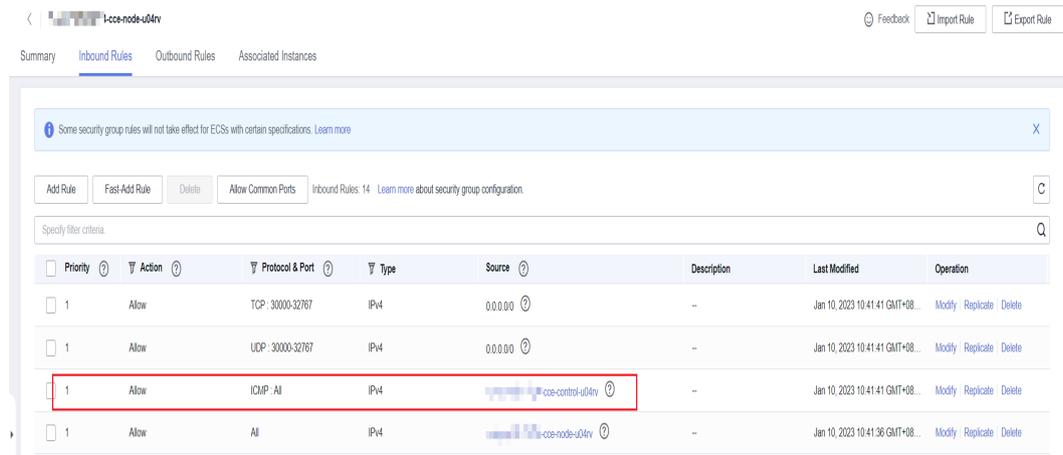
#### Solución

Inicie sesión en la consola de VPC, seleccione **Access Control > Security Groups** e introduzca el nombre del clúster de destino en el cuadro de búsqueda. Se muestran dos grupos de seguridad:

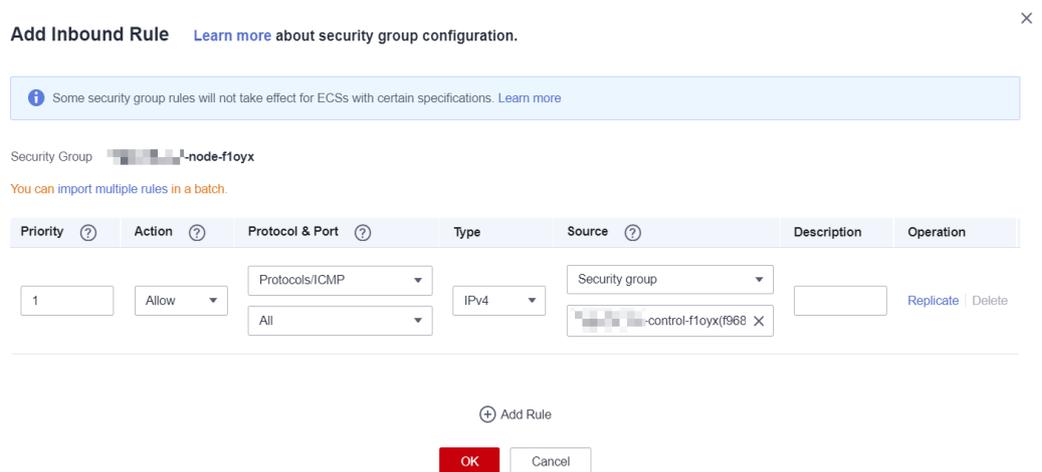
- El nombre del grupo de seguridad es **cluster name-node-xxx**. Este grupo de seguridad está asociado a los nodos de usuario.
- El nombre del grupo de seguridad es **cluster name-control-xxx**. Este grupo de seguridad está asociado con los nodos maestros.



Haga clic en el grupo de seguridad del usuario del nodo y asegúrese de que las siguientes reglas están configuradas para permitir que el nodo principal tenga acceso al nodo mediante **ICMP**.



De lo contrario, agregue una regla al grupo de seguridad del nodo. Establezca **Source** en **Security group** y descripción en "Created by CCE, please don't modify! Used by the master node to access the worker node."



## 2.5.7.9 Restricción del nodo de Arm

### Concepto de comprobación

Compruebe los siguientes aspectos:

- Compruebe si el clúster es un clúster Kunpeng o si el nodo principal de un clúster híbrido es de la arquitectura de ARM.
- Compruebe si el clúster contiene nodos de los nodos de Arm.

### Solución

- **Escenario 1: El clúster es un clúster de Kunpeng o el nodo principal de un clúster híbrido es de la arquitectura de Arm.**

La última versión del clúster de Kunpeng que se puede actualizar es v1.19. Si la versión que se va a actualizar es posterior a v1.19, no se puede realizar la actualización.

- **Escenario 2: El clúster contiene los nodos de Arm.**  
Eliminar los nodos de Arm.

### 2.5.7.10 Nodo por migrar

## Concepto de comprobación

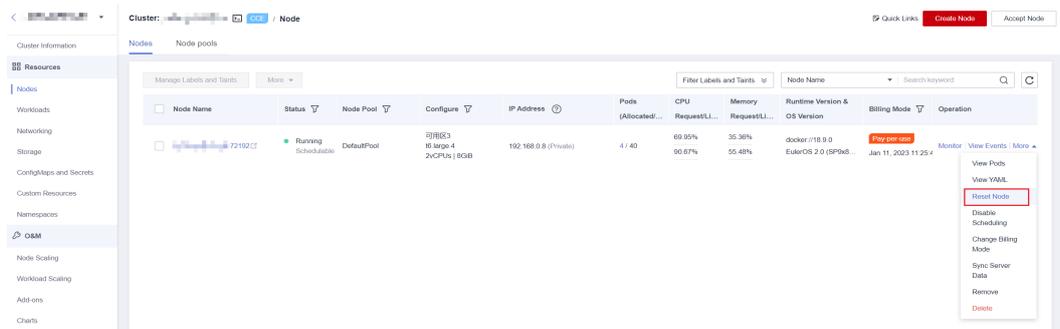
Compruebe si el nodo necesita ser migrado.

## Solución

Para el clúster 1.15 que se actualiza desde 1.13 en modo continuo, debe migrar (reiniciar o crear y reemplazar) todos los nodos antes de realizar la actualización de nuevo.

### Solución 1

Vaya a la consola de CCE y acceda a la consola del clúster. Elija **Nodes** en el panel de navegación y haga clic en **More > Reset Node** en la columna **Operation** del nodo correspondiente. Para obtener más información, véase [Restablecimiento de un nodo](#). Una vez restablecido el nodo, vuelva a intentar la tarea de comprobación.



### NOTA

Al restablecer un nodo se restablecerán todas las etiquetas de nodo, lo que puede afectar a la programación de la carga de trabajo. Antes de restablecer un nodo, compruebe y conserve las etiquetas que haya agregado manualmente al nodo.

### Solución 2

Después de crear un nodo, elimine el nodo defectuoso.

### 2.5.7.11 Recurso de Kubernetes descartado

## Concepto de comprobación

Compruebe si hay recursos descartados en los clústeres.

## Solución

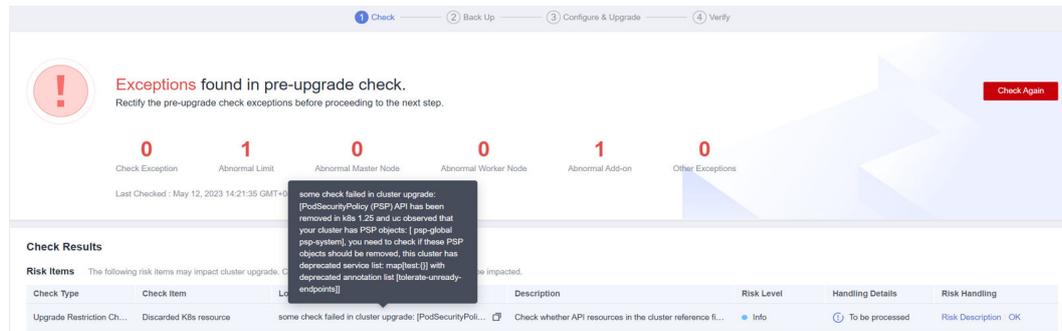
**Escenario 1: El objeto de recurso de PodSecurityPolicy se ha descartado desde los clústeres de 1.25.**



Ejecute el comando **kubectrl get psp -A** en el clúster para obtener el objeto PSP existente.

Si estos dos objetos no se utilizan, omita la comprobación. De lo contrario, actualice las funciones correspondientes a PodSecurity haciendo referencia a los documentos de [Seguridad del pod](#) relacionados.

**Escenario 2: El Service en los clústeres de 1.25 o posterior ha descartado la anotación: tolerate-unready-endpoints.**



Compruebe si el Service proporcionado en la información de log contiene la anotación de **tolerate-unready-endpoints**. En caso afirmativo, reemplace la anotación con los siguientes campos:

```
publishNotReadyAddresses: true
```

### 2.5.7.12 Riesgo de compatibilidad

#### Concepto de comprobación

Lea las diferencias de compatibilidad de versiones y asegúrese de que no se vean afectadas.

La actualización del parche no implica diferencias de compatibilidad de versiones.

## Compatibilidad de versiones

Ruta de actualización	Precaución	Autoverificación
v1.19 a v1.21 o v1.23	<p>El error de <b>exec probe timeouts</b> se corrige en Kubernetes 1.21. Antes de esta corrección de errores, la sonda exec no tiene en cuenta el campo <b>timeoutSeconds</b>. En su lugar, la sonda se ejecutará indefinidamente, incluso más allá de su fecha límite configurada. Se detendrá hasta que se devuelva el resultado. Si no se especifica este campo, se utiliza el valor predeterminado <b>1</b>. Este campo entra en vigor después de la actualización. Si el sondeo se ejecuta durante 1 segundo, la comprobación de estado de la aplicación puede fallar y la aplicación puede reiniciarse con frecuencia.</p>	<p>Antes de la actualización, compruebe si el tiempo de espera está configurado correctamente para la sonda exec.</p>
	<p>kube-apiserver de CCE 1.19 o posterior requiere que el campo Subject Alternative Names (SANs) esté configurado para el certificado de su servidor webhook. De lo contrario, kube-apiserver no puede invocar al servidor webhook después de la actualización, y contenedores no se puede iniciar correctamente.</p> <p>Causa raíz: X.509 <b>CommonName</b> se descarta en Go 1.15. kube-apiserver de CCE 1.19 se compila usando Go 1.15. Si su certificado webhook no tiene SAN, kube-apiserver no procesa el campo <b>CommonName</b> del certificado X.509 como nombre de host de forma predeterminada. Como resultado, la autenticación falla.</p>	<p>Antes de la actualización, compruebe si el campo SAN está configurado en el certificado de su servidor webhook.</p> <ul style="list-style-type: none"> <li>● Si no tiene su propio servidor webhook, puede omitir esta comprobación.</li> <li>● Si el campo no está definido, se recomienda utilizar el campo SAN para especificar la dirección IP y el nombre de dominio admitidos por el certificado.</li> </ul>
	<p>Los nodos de Arm no se admiten en los clústeres de v1.21 y posteriores.</p>	<p>Compruebe si sus servicios se verán afectados si no se pueden utilizar los nodos de Arm.</p>

Ruta de actualización	Precaución	Autoverificación
<p>v1.15 a v1.19</p>	<p>El plano de control de en los clústeres v1.19 es incompatible con kubelet v1.15. Si un nodo no se puede actualizar o el nodo que se va a actualizar se reinicia después de que el nodo principal se actualice con éxito, hay una alta probabilidad de que el nodo esté en el estado <b>NotReady</b>.</p> <p>Esto se debe a que el nodo no puede actualizarse reinicia el kubelet y activa el registro del nodo. En los clústeres de v1.15, las etiquetas de registro predeterminadas <b>failure-domain.beta.kubernetes.io/is-baremetal</b> y <b>kubernetes.io/availablezone</b> son consideradas como etiquetas no válidas por los clústeres de v1.19.</p> <p>Las etiquetas válidas en los clústeres de v1.19 son <b>node.kubernetes.io/baremetal</b> y <b>failure-domain.beta.kubernetes.io/zone</b>.</p>	<ol style="list-style-type: none"> <li>1. En los casos normales, este escenario no se activa.</li> <li>2. Después de actualizar el nodo principal, no suspenda la actualización para que el nodo pueda actualizarse rápidamente.</li> <li>3. Si un nodo no se puede actualizar y no se puede restaurar, desaloje las aplicaciones en el nodo tan pronto como sea posible. Póngase en contacto con el soporte técnico y omita la actualización del nodo. Una vez completada la actualización, restablezca el nodo.</li> </ol>

Ruta de actualización	Precaución	Autoverificación
	<p>En los clústeres 1.15 y 1.19 de CCE, el sistema de archivos del controlador de almacenamiento de Docker cambia de XFS a Ext4. Como resultado, la secuencia de paquetes de importación en los pods de la aplicación Java actualizada puede ser anormal, causando excepciones de pod.</p>	<p>Antes de la actualización, compruebe el archivo de configuración de Docker <code>/etc/docker/daemon.json</code> en el nodo. Compruebe si el valor de <code>dm.fs</code> es de <code>xfs</code>.</p> <ul style="list-style-type: none"> <li>● Si el valor es de <code>ext4</code> o el controlador de almacenamiento es Overlay, puede omitir los siguientes pasos.</li> <li>● Si el valor es de <code>xfs</code>, se recomienda desplegar aplicaciones en el clúster de la nueva versión con antelación para probar si las aplicaciones son compatibles con la nueva versión del clúster.</li> </ul> <pre data-bbox="975 898 1430 1227"> {   "storage-driver":   "devicemapper",   "storage-opts": [     "dm.thinpooldev=/dev/mapper/     vgpaas-thinpool",     "dm.use_deferred_removal=true",     "dm.fs=xfs",     "dm.use_deferred_deletion=true"   ] }                     </pre>
	<p>kube-apiserver de CCE 1.19 o posterior requiere que el campo Subject Alternative Names (SANs) esté configurado para el certificado de su servidor webhook. De lo contrario, kube-apiserver no puede invocar al servidor webhook después de la actualización, y contenedores no se puede iniciar correctamente.</p> <p>Causa raíz: X.509 <code>CommonName</code> se descarta en Go 1.15. kube-apiserver de CCE 1.19 se compila usando Go 1.15. El campo <code>CommonName</code> se procesa como el nombre de host. Como resultado, la autenticación falla.</p>	<p>Antes de la actualización, compruebe si el campo SAN está configurado en el certificado de su servidor webhook.</p> <ul style="list-style-type: none"> <li>● Si no tiene su propio servidor webhook, puede omitir esta comprobación.</li> <li>● Si el campo no está definido, se recomienda utilizar el campo SAN para especificar la dirección IP y el nombre de dominio admitidos por el certificado.</li> </ul> <p><b>AVISO</b></p> <p>Para mitigar el impacto de las diferencias de versión en la actualización del clúster, CCE realiza un procesamiento especial durante la actualización de 1.15 a 1.19 y sigue soportando certificados sin SAN. Sin embargo, no se requiere ningún procesamiento especial para las actualizaciones posteriores. Le aconsejamos que rectifique su certificado lo antes posible.</p>

Ruta de actualización	Precaución	Autoverificación
	En clústeres de v1.17.17 y posteriores, CCE crea automáticamente políticas de seguridad de pods (PSP) para usted, que restringen la creación de pods con configuraciones inseguras, por ejemplo, pods para los que <b>net.core.somaxconn</b> bajo un <code>sysctl</code> está configurado en el contexto de seguridad.	Después de una actualización, puede permitir configuraciones de sistema inseguras según sea necesario. Para obtener más información, véase <a href="#">Configuración de una política de seguridad de pod</a> .
v1.13 a v1.15	Después de actualizar un clúster de red de VPC, el nodo principal ocupa un bloque CIDR adicional debido a la actualización de los componentes de red. Si no hay ningún bloque CIDR contenedor disponible para el nuevo nodo, el pod programado para el nodo no puede ejecutarse.	Generalmente, este problema se produce cuando los nodos en el clúster están a punto de ocupar completamente el bloque CIDR contenedor. Por ejemplo, el bloque CIDR contenedor es 10.0.0.0/16, el número de direcciones IP disponibles es de 65,536 y a la red VPC se le asigna un bloque CIDR con el tamaño fijo (utilizando la máscara para determinar el número máximo de direcciones IP contenedor asignadas a cada nodo). Si el límite superior es 128, el clúster admite un máximo de 512 (65536/128) nodos, incluidos los tres nodos principales. Después de actualizar el clúster, cada uno de los tres nodos principales ocupa un bloque CIDR. Como resultado, se soportan 506 nodos.

### 2.5.7.13 Versión de nodo de CCEAgent

#### Concepto de comprobación

Compruebe si `cce-agent` en el nodo actual es de la versión más reciente.

#### Solución

Si `cce-agent` no es de la última versión, la actualización automática falla. Este problema suele ser causado por una dirección de OBS no válida o la versión del componente está desactualizada.

**Paso 1** Inicie sesión en un nodo normal que pasa la comprobación, obtenga la ruta del archivo de configuración `cce-agent` y compruebe la dirección de OBS.

```
cat `ps aux | grep cce-agent | grep -v grep | awk -F '-' '{print $2}'`
```



## Solución

- **Actualice el clúster durante las horas no pico.**
- Compruebe si hay demasiados pods desplegados en el nodo. Si es así, re programe los pods a otros nodos inactivos.

### 2.5.7.15 Comprobación de CRD

#### Concepto de comprobación

Compruebe los siguientes aspectos:

- Compruebe si se ha eliminado **packageversions.version.cce.io** de CRD clave del clúster.
- Compruebe si se ha eliminado **network-attachment-definitions.k8s.cni.cncf.io** de CRD de clave de clúster.

## Solución

Si los resultados de la comprobación son anormales, póngase en contacto con el soporte técnico.

### 2.5.7.16 Disco de nodo

#### Concepto de comprobación

Compruebe los siguientes aspectos:

- Compruebe si los discos de datos clave del nodo cumplen con los requisitos de actualización.
- Compruebe si el directorio **/tmp** tiene 500 MB de espacio disponible.

## Solución

Durante la actualización del nodo, los discos clave almacenan el paquete de componentes de actualización y el directorio **/tmp** almacena archivos temporales.

- **Escenario 1: Compruebe si el disco cumple con los requisitos de actualización.**  
Ejecute el siguiente comando para comprobar el uso de cada disco de clave. Después de asegurarse de que el espacio disponible cumple con los requisitos y comprobar de nuevo. Si el espacio del nodo principal es insuficiente, póngase en contacto con el soporte técnico.
  - Partición de disco de Docker: 2 GB para nodos principales y 1 GB para nodos de trabajo  

```
df -h /var/lib/docker
```
  - Partición de disco de containerd: 2 GB para nodos principales y 1 GB para nodos de trabajo  

```
df -h /var/lib/containerd
```
  - Partición de disco de kubelet: 2 GB para nodos principales y 1 GB para nodos de trabajo  

```
df -h /var/lib/docker
```
  - Disco de sistema: 10 GB para nodos principales y 2 GB para nodos de trabajo  

```
df -h /
```

- **Escenario 2: El espacio de directorio /tmp es insuficiente.**

Ejecute el siguiente comando para comprobar el uso del sistema de archivos donde se encuentra el directorio **/tmp**. Asegúrese de que el espacio es superior a 500 MB y vuelva a comprobarlo.

```
df -h /tmp
```

### 2.5.7.17 Nodo de DNS

#### Concepto de comprobación

Compruebe los siguientes aspectos:

- Compruebe si la configuración de DNS del nodo actual puede resolver la dirección de OBS.
- Compruebe si el nodo actual puede acceder a la dirección de OBS del paquete de componentes de actualización de almacenamiento.

#### Solución

Durante la actualización del nodo, debe obtener el paquete de componentes de actualización de OBS. Si esta comprobación falla, póngase en contacto con el soporte técnico.

### 2.5.7.18 Permisos de archivo de directorio de clave de nodo

#### Concepto de comprobación

Compruebe si el directorio de claves **/var/paas** en los nodos contiene archivos con propietarios o grupos de propietarios anormales.

#### Solución

CCE utiliza el directorio **/var/paas** para gestionar los nodos y almacenar datos de archivos cuyo grupo de propietarios y propietario son paas.

Durante la actualización del clúster actual, el grupo de propietarios y propietario de los archivos en el directorio **/var/paas** se restablecen a paas.

Compruebe si los datos del archivo están almacenados en el directorio **/var/paas**. Si es así, no utilice este directorio, quite los archivos anormales de este directorio y vuelva a comprobarlo. De lo contrario, la actualización está prohibida.

### 2.5.7.19 Kubelet

#### Concepto de comprobación

Compruebe si el kubelet del nodo se está ejecutando correctamente.

#### Solución

- **Escenario 1: El estado de kubelet es anormal.**

Si el kubelet es anormal, el nodo no está disponible. Restaure el nodo siguiendo las instrucciones de [¿Qué debo hacer si un clúster está disponible pero algunos nodos no están disponibles?](#) y vuelva a comprobarlo.

- **Escenario 2: La versión de cce-pause version es anormal.**

La versión de la imagen de contenedor de pausa de la que depende kubelet no es cce-pause:3.1. Si continúa con la actualización, los pods se reiniciarán por lotes. Actualmente, la actualización no es compatible. Contacte con el servicio de asistencia técnica.

## 2.5.7.20 Memoria de nodos

### Concepto de comprobación

Compruebe si el uso de memoria del nodo supera el 90%.

### Solución

- **Actualice el clúster durante las horas no pico.**
- Compruebe si hay demasiados pods desplegados en el nodo. Si es así, re programe los pods a otros nodos inactivos.

## 2.5.7.21 Servidor de sincronización de reloj de nodo

### Concepto de comprobación

Compruebe si el servidor de sincronización de reloj ntpd o chronyd del nodo se está ejecutando correctamente.

### Solución

- **Escenario 1: ntpd se ejecuta anormalmente.**

Inicie sesión en el nodo y ejecute el comando `systemctl status ntpd` para consultar el estado de ejecución de ntpd. Si el resultado del comando es anormal, ejecute el comando `systemctl restart ntpd` y vuelva a consultar el estado.

El resultado del comando normal es el siguiente:

```
[root@xxxxxxxxx paas]# systemctl status ntpd
● ntpd.service - Network Time Service
   Loaded: loaded (/usr/lib/systemd/system/ntpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2022-12-06 14:52:30 CST; 4 days ago
     Main PID: 8587 (ntpd)
        Tasks: 2
       Memory: 1.6M
      CGroup: /system.slice/ntpd.service
             └─8587 /usr/sbin/ntpd -u ntp:ntp -g -x
```

Si el problema persiste después de reiniciar ntpd, póngase en contacto con el soporte técnico.

- **Escenario 2: chronyd se ejecuta anormalmente.**

Inicie sesión en el nodo y ejecute el comando `systemctl status chronyd` para consultar el estado de ejecución de chronyd. Si el resultado del comando es anormal, ejecute el comando `systemctl restart chronyd` y vuelva a consultar el estado.

El resultado del comando normal es el siguiente:

```
root@xxxxxxxxxxxxxxxxx:~# systemctl status chronyd
● chrony.service - chrony, an NTP client/server
   Loaded: loaded (/lib/systemd/system/chrony.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2022-08-24 16:33:28 CST; 3 months 16 days ago
     Docs: man:chronyd(8)
           man:chronyc(1)
           man:chrony.conf(5)
   Process: 6492 ExecStartPost=/usr/lib/chrony/chrony-helper update-daemon (code=exited, status=0/SUCCESS)
   Process: 6461 ExecStart=/usr/lib/systemd/scripts/chronyd-starter.sh $DAEMON_OPTS (code=exited, status=0/SUCCESS)
  Main PID: 6488 (chronyd)
    Tasks: 1 (limit: 4915)
   CGroup: /system.slice/chrony.service
           └─6488 /usr/sbin/chronyd
```

Si el problema persiste después de reiniciar chronyd, póngase en contacto con el soporte técnico.

## 2.5.7.22 SO del nodo

### Concepto de comprobación

Compruebe si la versión del kernel del sistema operativo del nodo es compatible con CCE.

### Solución

Los nodos en ejecución dependen de la versión estándar inicial del núcleo cuando se crean. CCE ha realizado pruebas de compatibilidad exhaustivas basadas en esta versión del kernel. Una versión no estándar del núcleo puede causar problemas de compatibilidad inesperados durante la actualización del nodo y la actualización del nodo puede fallar. Para obtener más información, véase [Operaciones y soluciones de alto riesgo](#).

Actualmente, este tipo de nodos no deben actualizarse. Se recomienda restablecer el nodo a la versión estándar del núcleo antes de la actualización siguiendo las instrucciones de [Restablecimiento de un nodo](#).

## 2.5.7.23 Recuento de CPU de nodo

### Concepto de comprobación

Compruebe si el número de CPUs en el nodo principal es mayor que 2.

### Solución

Si el número de las CPU en el nodo principal es 2, póngase en contacto con el soporte técnico para ampliar el número a 4 o más.

## 2.5.7.24 Comando de nodo de Python

### Concepto de comprobación

Compruebe si los comandos de Python están disponibles en un nodo.

### Método de comprobación

```
/usr/bin/python --version
echo $?
```

Si el resultado del comando no es 0, la comprobación falla.

## Solución

Instale Python antes de la actualización.

### 2.5.7.25 Versión de ASM

#### Concepto de comprobación

Compruebe los siguientes aspectos:

- Compruebe si el clúster utiliza ASM.
- Compruebe si la versión actual de ASM admite la versión del clúster de destino.

## Solución

- Actualice ASM y, a continuación, actualice el clúster. Las reglas de adaptación entre las versiones de ASM y clúster son las siguientes:

**Tabla 2-21** Reglas de adaptación entre las versiones de ASM y clúster

Versión de ASM	Versión del clúster
1.3	v1.13/15/17/19.*
1.6	v1.15/17/19/21.*
1.8	v1.15/17/19/21.*
1.13	v1.21/23.*

- Si no se requiere ASM, elimínelo antes de la actualización. Después de la actualización, el clúster no se puede vincular a ASM que no coincida con la tabla. Por ejemplo, si desea actualizar un clúster de v1.21 y ASM de v1.8 a v1.23, actualice el ASM primero.

### 2.5.7.26 Preparación del nodo

#### Concepto de comprobación

Compruebe si los nodos del clúster están listos.

## Solución

- **Escenario 1: Los nodos están en el estado no disponible.**  
 Inicie sesión en la consola de CCE y acceda a la consola del clúster. Elija **Nodos** en el panel de navegación y filtre los nodos no disponibles, rectifique los nodos defectuosos haciendo referencia a las sugerencias proporcionadas por la consola y vuelva a comprobarlo.
- **Escenario 2: El estado del nodo mostrado es incompatible con el estado real.**  
 Las causas posibles son:
  - a. El estado del nodo es normal en la página de nodos, pero el resultado de la comprobación muestra que el nodo no está listo. Compruébelo de nuevo.

- b. El nodo no se encuentra en la página de nodos, pero el resultado de la comprobación muestra que el nodo está en el clúster. Contacte con el servicio de asistencia técnica.

### 2.5.7.27 Diario de nodo

#### Concepto de comprobación

Compruebe si el diario de un nodo es normal.

#### Solución

Inicie sesión en el nodo y ejecute el comando **systemctl is-active systemd-journald** para consultar el estado de ejecución de diario. Si el resultado del comando es anormal, ejecute el comando **systemctl restart systemd-journald** y vuelva a consultar el estado.

El resultado del comando normal es el siguiente:

```
[root@xxxxxxxxxxxxx paas]# systemctl is-active systemd-journald
active
```

Si el problema persiste después de que se reinicie el diario, póngase en contacto con el soporte técnico.

### 2.5.7.28 containerd.sock

#### Concepto de comprobación

Compruebe si el archivo `containerd.sock` existe en el nodo. Este archivo afecta al inicio del tiempo de ejecución de contenedor en el Euler OS.

#### Solución

**Escenario: El Docker utilizado por el nodo es el Euler-docker personalizado.**

**Paso 1** Inicie sesión en el nodo.

**Paso 2** Ejecute el comando **rpm -qa | grep docker | grep euleros**. Si la salida del comando no está vacía, el Docker utilizado en el nodo es Euler-docker.

**Paso 3** Ejecute el comando **ls /run/containerd/containerd.sock**. Si el archivo existe, Docker no puede iniciarse.

**Paso 4** Ejecute el comando **rm -rf /run/containerd/containerd.sock** y vuelva a realizar la comprobación de actualización del clúster.

----Fin

### 2.5.7.29 Error interno

#### Concepto de comprobación

Antes de la actualización, compruebe si se produce un error interno.

## Solución

Si esta comprobación falla, póngase en contacto con el soporte técnico.

### 2.5.7.30 Punto de montaje del nodo

#### Concepto de comprobación

Compruebe si existen puntos de montaje inaccesibles en el nodo.

## Solución

**Escenario: hay puntos de montaje inaccesibles en el nodo.**

Si el nodo utiliza NFS de red (como OBS, SFS y SFS) y el nodo se desconecta con el servidor NFS, el punto de montaje sería inaccesible y todos los procesos que acceden a este punto de montaje se suspenden.

**Paso 1** Inicie sesión en el nodo.

**Paso 2** Ejecute los siguientes comandos en el nodo en secuencia:

```
- df -h  
- for dir in `df -h | grep -v "Mounted on" | awk "{print \\$NF}"`;do cd $dir;  
done && echo "ok"
```

**Paso 3** Si se devuelve **ok**, no se produce ningún problema.

De lo contrario, inicie otro terminal y ejecute el siguiente comando para comprobar si el comando anterior está en el estado D:

```
- ps aux | grep "D "
```

**Paso 4** Si un proceso está en el estado D, se produce el problema. Solo puede restablecer el nodo para resolver el problema. Reinicie el nodo y actualice el clúster de nuevo. Para obtener más información acerca de cómo restablecer un nodo, consulte [Restablecimiento de un nodo](#).

#### **NOTA**

Al restablecer un nodo se restablecerán todas las etiquetas de nodo, lo que puede afectar a la programación de la carga de trabajo. Antes de restablecer un nodo, compruebe y conserve las etiquetas que haya agregado manualmente al nodo.

----Fin

### 2.5.7.31 Mancha de nodo de Kubernetes

#### Concepto de comprobación

Compruebe si la mancha, como se muestra en la [Tabla 2-22](#), existe en el nodo.

**Tabla 2-22** Lista de comprobación de manchas

Nombre	Impacto
node.kubernetes.io/upgrade	NoSchedule

## Solución

Escenario 1: El nodo se omite durante la actualización del clúster.

**Paso 1** Para obtener más información sobre cómo configurar kubectl, consulte [Conexión a un clúster con kubectl](#).

**Paso 2** Compruebe la versión de kubelet del nodo correspondiente. Si se espera la siguiente información:

```
[root@10-3-120-59 paas]# kubectl get node
NAME              STATUS    ROLES    AGE    VERSION
10.3.5[redacted] Ready     <none>   28h    v1.19.16-r4-CCE22.11.1
10.3.5[redacted] Ready     <none>   28h    v1.19.16-r4-CCE22.11.1
```

Si la versión del nodo es diferente de la de otros nodos, el nodo se omite durante la actualización. Reinicie el nodo y actualice el clúster de nuevo. Para obtener más información acerca de cómo restablecer un nodo, consulte [Restablecimiento de un nodo](#).

### 📖 NOTA

Al restablecer un nodo se restablecerán todas las etiquetas de nodo, lo que puede afectar a la programación de la carga de trabajo. Antes de restablecer un nodo, compruebe y conserve las etiquetas que haya agregado manualmente al nodo.

---Fin

## 2.5.7.32 Restricción de everest

### Concepto de comprobación

Comprueba si el complemento de everest actual tiene las restricciones de compatibilidad. Consulte [Tabla 2-23](#).

**Tabla 2-23** Lista de versiones adicionales de everest que tienen las restricciones de compatibilidad

Nombre del complemento	Versiones involucradas
everest	v1.0.2-v1.0.7 v1.1.1-v1.1.5

## Solución

El complemento de everest actual tiene restricciones de compatibilidad y no se puede actualizar con la actualización del clúster. Contacte con el servicio de asistencia técnica.

### 2.5.7.33 Restricción de cce-hpa-controller

#### Concepto de comprobación

Compruebe si el complemento de cce-controller-hpa actual tiene restricciones de compatibilidad.

#### Solución

El complemento de cce-controller-hpa actual tiene restricciones de compatibilidad. Se debe instalar en el clúster un complemento que pueda proporcionar API de métricas, por ejemplo, servidor de métricas.

### 2.5.7.34 Enlace mejorado del núcleo de la CPU

#### Concepto de comprobación

Compruebe si la versión actual del clúster y la versión de destino admiten un [enlace mejorado del núcleo de la CPU](#).

#### Solución

**Escenario:** Solo la versión de clúster actual admite un enlace mejorado del núcleo de la CPU.

Actualice a una versión de clúster que admita un enlace mejorado del núcleo de CPU. En la siguiente tabla se enumeran las versiones de clúster que admiten la vinculación mejorada del núcleo de CPU.

**Tabla 2-24** Lista de versiones de clúster que admiten un enlace mejorado del núcleo de CPU

Versión del clúster	Enlace mejorado del núcleo de la CPU
Clústeres de v1.17 o anteriores	No se admite
Clústeres de v1.19	No se admite
Clústeres de v1.21	No se admite
Clústeres de v1.23 y posteriores	Se admite

### 2.5.7.35 Estado de componentes de nodo de usuario

#### Concepto de comprobación

Compruebe si el tiempo de ejecución contenedor y los componentes de red en el nodo de usuario están en buen estado.

#### Solución

Si un componente es anormal, inicie sesión en el nodo para comprobar el estado del componente anormal y rectificar la falla.

### 2.5.7.36 Estado de los componentes del nodo del controlador

#### Concepto de comprobación

Compruebe si los componentes de Kubernetes, tiempo de ejecución de contenedor y red del nodo del controlador están en buen estado.

#### Solución

Si un componente en el nodo del controlador es anormal, póngase en contacto con el soporte técnico.

### 2.5.7.37 Límite de recursos de memoria del componente de Kubernetes

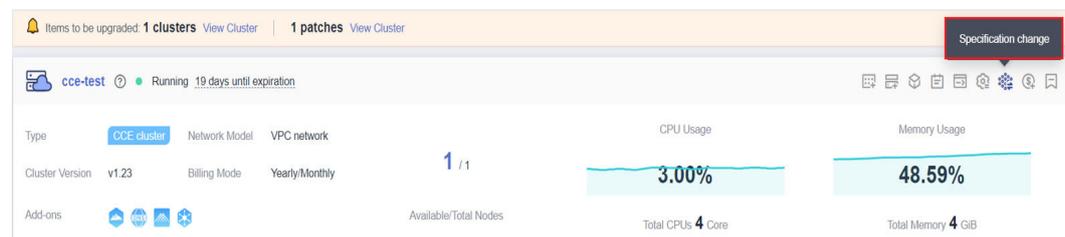
#### Concepto de comprobación

Compruebe si los recursos de los componentes de Kubernetes, como etcd y kube-controller-manager, exceden el límite superior.

#### Solución

Solución 1: Reduzca los recursos de Kubernetes.

Solución 2: [Cambio de escala de clúster](#)



### 2.5.7.38 API de Kubernetes descartadas

#### Concepto de comprobación

Compruebe si la API invocada se ha descartado en la versión de Kubernetes de destino.

#### Solución

Basado en el resultado de la comprobación, verifique los detalles sobre la API invocada en el servicio. Una vez completada la actualización, rectifique la API descartada según [Notas del lanzamiento de Kubernetes](#) del [documento de la comunidad de Kubernetes](#).

### 2.5.7.39 Capacidad de IPv6 de un clúster de CCE Turbo

#### Concepto de comprobación

Si IPv6 está habilitado para un clúster de CCE Turbo, compruebe si la versión del clúster de destino admite IPv6.

## Solución

Los clústeres de CCE Turbo admiten IPv6 desde la v1.23. Esta función está disponible en las siguientes versiones:

- v1.23: 1.23.8-r0 o posterior
- v1.25: 1.25.3-r0 o posterior
- v1.25 o posterior

Si IPv6 se ha habilitado en el clúster antes de la actualización, la versión del clúster de destino también debe admitir IPv6. Seleccione una versión de clúster adecuada.

### 2.5.7.40 NetworkManager de nodo

#### Concepto de comprobación

Compruebe si el NetworkManager de un nodo es normal.

#### Solución

Inicie sesión en el nodo y ejecute el comando `systemctl is-active NetworkManager` para consultar el estado de ejecución de NetworkManager. Si el resultado del comando es anormal, ejecute el comando `systemctl restart NetworkManager` y vuelva a consultar el estado.

Si el problema persiste después de reiniciar NetworkManager, póngase en contacto con el soporte técnico.

### 2.5.7.41 Archivo de ID de nodo

#### Concepto de comprobación

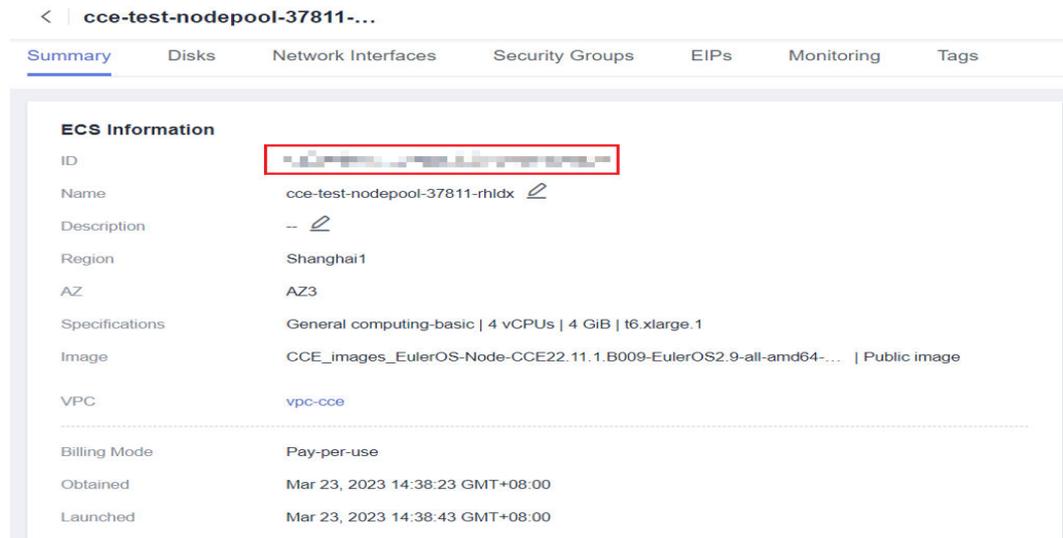
Compruebe el formato de archivo ID.

#### Solución

**Paso 1** En la página **Nodes** de la consola de CCE, haga clic en el nombre del nodo anormal para ir a la página de ECS.



**Paso 2** Copie el ID de nodo y guárdelo en el host local.



**Paso 3** Inicie sesión en el nodo anormal y haga una copia de respaldo de los archivos.

```
cp /var/lib/cloud/data/instance-id /tmp/instance-id
cp /var/paas/conf/server.conf /tmp/server.conf
```

**Paso 4** Inicie sesión en el nodo anormal y escriba el ID de nodo obtenido en el archivo.

```
echo "Node ID" >> /var/lib/cloud/data/instance-id
echo "Node ID" >> /var/paas/conf/server.conf
```

----Fin

## 2.5.7.42 Consistencia de la configuración del nodo

### Concepto de comprobación

Al actualizar un clúster de CCE a v1.19 o posterior, el sistema comprueba si se han modificado los siguientes archivos de configuración en segundo plano:

- /opt/cloud/cce/kubernetes/kubelet/kubelet
- /opt/cloud/cce/kubernetes/kubelet/kubelet\_config.yaml
- /opt/cloud/cce/kubernetes/kube-proxy/kube-proxy
- /etc/containerd/default\_runtime\_spec.json
- /etc/sysconfig/docker
- /etc/default/docker
- /etc/docker/daemon.json

Si modifica algunos parámetros en estos archivos, la actualización del clúster puede fallar o los servicios pueden ser anormales después de la actualización. Si confirma que la modificación no afecta a los servicios, continúe con la actualización.

#### NOTA

CCE utiliza el script de imagen estándar para comprobar la consistencia de la configuración del nodo. Si utiliza otras imágenes personalizadas, la comprobación puede fallar.

La modificación esperada no será interceptada. En la siguiente tabla se enumeran los parámetros que se pueden modificar.

**Tabla 2-25** Parámetros que se pueden modificar

Compo nente	Archivo de configuración	Parámetro	Versión actualizad a
kubelet	/opt/cloud/cce/kubernetes/kubelet/kubelet_config.yaml	cpuManagerPolicy	Más tarde que v1.19
kubelet	/opt/cloud/cce/kubernetes/kubelet/kubelet_config.yaml	maxPods	Más tarde que v1.19
kubelet	/opt/cloud/cce/kubernetes/kubelet/kubelet_config.yaml	kubeAPIQPS	Más tarde que v1.19
kubelet	/opt/cloud/cce/kubernetes/kubelet/kubelet_config.yaml	kubeAPIBurst	Más tarde que v1.19
kubelet	/opt/cloud/cce/kubernetes/kubelet/kubelet_config.yaml	podPidsLimit	Más tarde que v1.19
kubelet	/opt/cloud/cce/kubernetes/kubelet/kubelet_config.yaml	topologyManager-Policy	Más tarde que v1.19
kubelet	/opt/cloud/cce/kubernetes/kubelet/kubelet_config.yaml	resolvConf	Más tarde que v1.19
kubelet	/opt/cloud/cce/kubernetes/kubelet/kubelet_config.yaml	eventRecordQPS	Más tarde que v1.21
kubelet	/opt/cloud/cce/kubernetes/kubelet/kubelet_config.yaml	topologyManager-Scope	Más tarde que v1.21
kubelet	/opt/cloud/cce/kubernetes/kubelet/kubelet_config.yaml	allowedUnsafeSysctl s	Más tarde que v1.19
docker	/etc/docker/daemon.json	dm.basesize	Más tarde que v1.19

## Solución

Si modifica algunos parámetros en estos archivos, pueden producirse excepciones después de la actualización. Si no está seguro de si los parámetros modificados afectarán a la actualización, póngase en contacto con el soporte técnico.

### 2.5.7.43 Archivo de configuración de nodo

#### Concepto de comprobación

Compruebe si los archivos de configuración de los componentes clave existen en el nodo.

En la siguiente tabla se enumeran los archivos que se van a comprobar.

Nombre del archivo	Contenido del archivo	Notas
/opt/cloud/cce/kubernetes/kubelet/kubelet	Parámetros de inicio de la línea de comandos de kubelet	-
/opt/cloud/cce/kubernetes/kubelet/kubelet_config.yaml	Parámetros de inicio de kubelet	-
/opt/cloud/cce/kubernetes/kube-proxy/kube-proxy	Parámetros de inicio de línea de comandos de kube-proxy	-
/etc/sysconfig/docker	Archivo de configuración de Docker	No se ha comprobado cuando se utiliza containerd o la máquina de Debain-Group.
/etc/default/docker	Archivo de configuración de Docker	No se marca cuando se utiliza containerd o la máquina de Centos-Group.

## Solución

Póngase en contacto con el soporte técnico para restaurar el archivo de configuración y, a continuación, realice la actualización.

### 2.5.7.44 Consistencia de la configuración de CoreDNS

#### Concepto de comprobación

Compruebe si la configuración actual de la clave de CoreDNS Corefile es diferente del registro de lanzamiento de Helm. La diferencia puede sobrescribirse durante la actualización del complemento, **afecta la resolución de nombres de dominio en el clúster**.

## Solución

Puede actualizar el complemento de coredns por separado después de confirmar las diferencias de configuración.

**Paso 1** Configure kubectl, consulte [Conexión a un clúster con kubectl](#).

**Paso 2** Obtenga el Corefile que entra en vigor actualmente.

```
kubectl get cm -nkube-system coredns -o jsonpath='{.data.Corefile}' >
corefile_now.txt
cat corefile_now.txt
```

**Paso 3** Obtenga el Corefile en el registro de lanzamiento de Helm (dependiendo de Python 3).

```
latest_release=`kubectl get secret -nkube-system -l owner=helm -l name=cceaddon-
coredns --sort-by=.metadata.creationTimestamp | awk 'END{print $1}'`
kubectl get secret -nkube-system $latest_release -o jsonpath='{.data.release}' |
base64 -d | base64 -d | gzip -d | python -m json.tool | python -c "
import json,sys,re,yaml;
```

```
manifests = json.load(sys.stdin)['manifest']
files = re.split('(?:^|\s*\n)---\s*',manifests)
for file in files:
    if 'coredns/templates/configmap.yaml' in file and 'Corefile' in file:
        corefile = yaml.safe_load(file)['data']['Corefile']
        print(corefile,end='')
        exit(0);
print('error')
exit(1);
" > corefile_record.txt
cat corefile_record.txt
```

**Paso 4** Compare las diferencias de salida entre **Paso 2** y **Paso 3**.

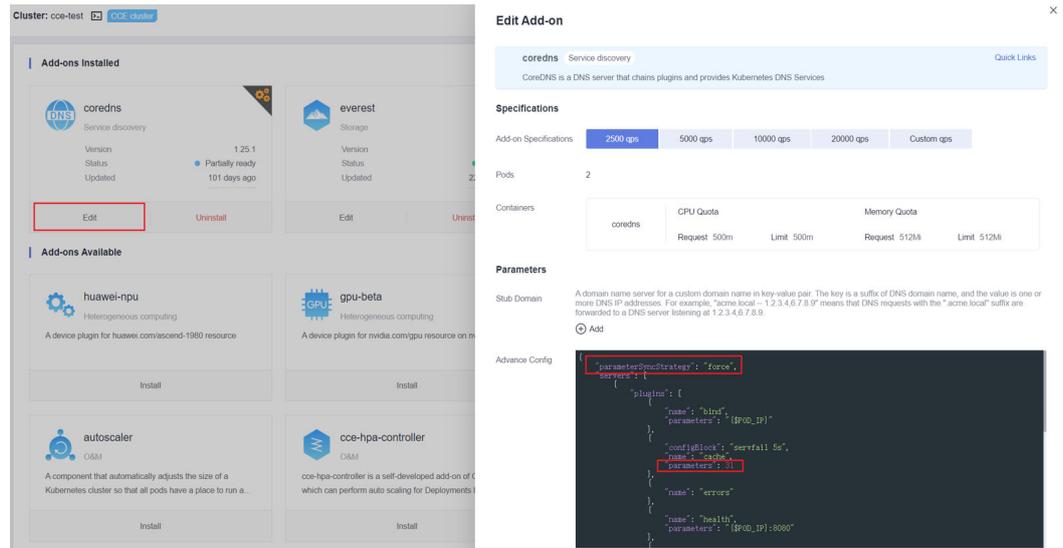
```
diff corefile_now.txt corefile_record.txt -y;
```

```
[root@... | paas]# diff corefile_now.txt corefile_record.tx
.:5353 {
  bind {$POD IP}
  cache 31
  errors
  health {$POD_IP}:8080
  kubernetes cluster.local in-addr.arpa ip6.arpa {
    pods insecure
    upstream /etc/resolv.conf
    fallthrough in-addr.arpa ip6.arpa
  }
  loadbalance round_robin
  prometheus {$POD_IP}:9153
  forward . /etc/resolv.conf
  reload
}
```

**Paso 5** Vuelva a la consola de CCE y haga clic en el nombre del clúster para ir a la consola del clúster. En la página **Add-ons**, seleccione el complemento de coredns y haga clic en **Upgrade**.

Para conservar las diferentes configuraciones, utilice cualquiera de los métodos siguientes:

- Establezca **parameterSyncStrategy** a **force**. Es necesario introducir manualmente la configuración diferencial. Para obtener más información, véase **coredns (complemento de recursos del sistema, obligatorio)**.
- Si **parameterSyncStrategy** se establece en **inherit**, las configuraciones diferenciadas se heredan automáticamente. El sistema analiza, identifica y hereda automáticamente parámetros diferenciados.



**Paso 6** Haga clic en **OK**. Una vez completada la actualización del complemento, compruebe si todas las instancias de coredns están disponibles y si Corefile cumple con las expectativas.

```
kubectl get cm -nkube-system coredns -o jsonpath='{.data.Corefile}'
```

**Paso 7** Cambie el valor de **parameterSyncStrategy** a **ensureConsistent** para habilitar la verificación de consistencia de la configuración.

Además, se recomienda utilizar la función de configuración de parámetros de la gestión de complementos de CCE para modificar la configuración de Corefile para evitar diferencias.

----Fin

## 2.6 Managing a Cluster

### 2.6.1 Gestión de configuración de clúster

#### Escenario

CCE le permite gestionar parámetros de clúster, con los cuales puede dejar que los componentes principales funcionen según sus requisitos.

#### Restricciones

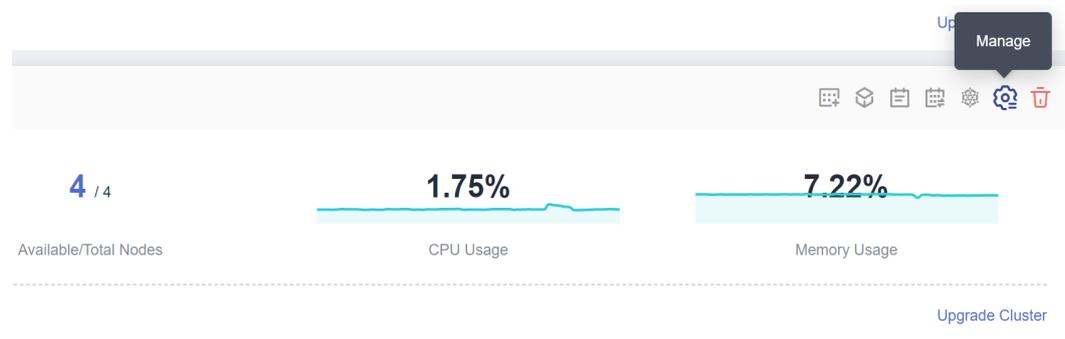
Esta función solo se admite en los clústeres de **v1.15 y posterior**. No se muestra para las versiones anteriores a la v1.15.

#### Procedimiento

**Paso 1** Inicie sesión en la consola de CCE. En el panel de navegación, elija **Clusters**.

**Paso 2** Haga clic en  junto al clúster de destino.

**Figura 2-14** Configuración



**Paso 3** En la página **Manage Component** de la derecha, cambie los valores de los siguientes parámetros de Kubernetes:

**Tabla 2-26** Parámetros de configuración del controlador extendido (soportados solo por clústeres de v1.21 y posteriores)

Parámetro	Descripción	Valor
enable-resource-quota	Si se crea automáticamente un objeto de cuota de recursos al crear un espacio de nombres. <ul style="list-style-type: none"> <li>● <b>false</b>: sin creación automática</li> <li>● <b>true</b>: creación automática habilitada. Para obtener más información sobre los valores predeterminados de la cuota de recursos, consulte <a href="#">Establecimiento de una cuota de recursos</a>.</li> </ul>	Predeterminado: false

**Tabla 2-27** Parámetros de kube-apiserver

Parámetro	Descripción	Valor
default-not-ready-toleration-seconds	Tiempo de tolerancia notReady, en segundos. NoExecute que se agrega de forma predeterminada a cada pod que no tiene tal tolerancia.	Predeterminado: 300s
default-unreachable-toleration-seconds	Tiempo de tolerancia inalcanzable, en segundos. NoExecute que se agrega de forma predeterminada a cada pod que no tiene tal tolerancia.	Predeterminado: 300s

Parámetro	Descripción	Valor
max-mutating-requests-inflight	<p>Número máximo de solicitudes de mutación simultáneas. Cuando se excede el valor de este parámetro, el servidor rechaza las solicitudes.</p> <p>El valor <b>0</b> no indica ninguna limitación.</p>	<p>La configuración manual ya no es compatible desde cluster v1.21. El valor se especifica automáticamente en función de la escala del clúster.</p> <ul style="list-style-type: none"> <li>● <b>200</b> para clústeres con 50 o 200 nodos</li> <li>● <b>500</b> para clústeres con 1,000 nodos</li> <li>● <b>1000</b> para clústeres con 2,000 nodos</li> </ul>
max-requests-inflight	<p>Número máximo de solicitudes simultáneas que no se muten. Cuando se excede el valor de este parámetro, el servidor rechaza las solicitudes.</p> <p>El valor <b>0</b> no indica ninguna limitación.</p>	<p>La configuración manual ya no es compatible desde cluster v1.21. El valor se especifica automáticamente en función de la escala del clúster.</p> <ul style="list-style-type: none"> <li>● <b>400</b> para clústeres con 50 o 200 nodos</li> <li>● <b>1000</b> para clústeres con 1,000 nodos</li> <li>● <b>2000</b> para clústeres con 2,000 nodos</li> </ul>

Parámetro	Descripción	Valor
service-node-port-range	Rango de puertos NodePort. Después de cambiar el valor, debe ir a la página de grupo de seguridad para cambiar el rango de puertos TCP/UDP de los grupos de seguridad de nodo 30000 a 32767. De lo contrario, los sistemas externos no pueden acceder a puertos distintos del puerto predeterminado.	Predeterminado: 30000-32767 Opciones: min>20105 max<32768
support-overload	Control de sobrecarga de clúster. Si está habilitado, las solicitudes simultáneas se controlan dinámicamente en función de la presión de recursos de los nodos maestros para mantenerlas disponibles y el clúster.  Este parámetro solo es compatible con clústeres de v1.23 o posterior.	<ul style="list-style-type: none"> <li>● false: El control de sobrecarga está deshabilitado.</li> <li>● true: El control de sobrecarga está habilitado.</li> </ul>

**Tabla 2-28** Parámetros de kube-controller-manager

Parámetro	Descripción	Valor
concurrent-deployment-syncs	Número de Deployments que se permiten sincronizar simultáneamente.	Predeterminado: 5
concurrent-endpoint-syncs	Número de puntos de conexión que se permiten sincronizar simultáneamente.	Predeterminado: 5
concurrent-gc-syncs	Número de trabajadores del recolector de basura a los que se permite sincronizar simultáneamente.	Predeterminado: 20
concurrent-job-syncs	Número de trabajos que se pueden sincronizar al mismo tiempo.	Predeterminado: 5
concurrent-namespace-syncs	Número de espacios de nombres que se permiten sincronizar simultáneamente.	Predeterminado: 10
concurrent-replicaset-syncs	Número de ReplicaSets que pueden sincronizarse simultáneamente.	Predeterminado: 5
concurrent-resource-quota-syncs	Número de cuotas de recursos que se permiten sincronizar simultáneamente.	Predeterminado: 5
concurrent-service-syncs	Número de servicios que se permiten sincronizar simultáneamente.	Predeterminado: 10

Parámetro	Descripción	Valor
concurrent-serviceaccount-token-syncs	Número de tokens de cuenta de servicio que se permiten sincronizar simultáneamente.	Predeterminado: 5
concurrent-ttl-after-finished-syncs	Número de trabajadores de controlador TTL-after-finished a los que se permite sincronizar simultáneamente.	Predeterminado: 5
concurrent_rc_syncs	Número de controladores de replicación que se permiten sincronizar simultáneamente.  <b>NOTA</b> Este parámetro se utiliza solo en clústeres de v1.19 o anteriores.	Predeterminado: 5
concurrent-rc-syncs	Número de controladores de replicación que se permiten sincronizar simultáneamente.  <b>NOTA</b> Este parámetro solo se utiliza en clústeres de versiones de v1.21 a v1.23. En versiones v1.25 y posteriores, este parámetro está obsoleto. (Este parámetro está obsoleto desde v1.25.3-r0.)	Predeterminado: 5
horizontal-pod-autoscaler-sync-period	Con qué frecuencia HPA audita métricas en un clúster.	Predeterminado: 15 segundos
kube-api-qps	Consulta por segundo (QPS) para usar mientras se habla con kube-apiserver.	Predeterminado: 100
kube-api-burst	Ráfaga para usar mientras se habla con kube-apiserver.	Predeterminado: 100
terminated-pod-gc-threshold	Número de pods terminados que pueden existir antes de que el recolector de basura de pods terminado comience a eliminar los pods terminados.  Si $\leq 0$ , el recolector de basura de pod terminado está deshabilitado.	Predeterminado: 1000

**Tabla 2-29** Parámetros de kube-scheduler

Parámetro	Descripción	Valor
kube-api-qps	Consulta por segundo (QPS) para usar mientras se habla con kube-apiserver.	Predeterminado: 100
kube-api-burst	Ráfaga para usar mientras se habla con kube-apiserver.	Predeterminado: 100

**Tabla 2-30** Parámetros eni (soportados solo por clústeres de CCE Turbo)

Parámetro	Descripción	Valor
nic-minimum-target	Número mínimo de ENI vinculados a un nodo a nivel de clúster	Predeterminado: 10
nic-maximum-target	Número máximo de ENI preenzaladas a un nodo a nivel de clúster	Predeterminado: 0
nic-warm-target	Número de ENI preenzaladas a un nodo a nivel de clúster	Predeterminado: 2
nic-max-above-warm-target	Recupere el número de ENI preenzaladas a un nodo a nivel de clúster	Predeterminado: 2
prebound-subeni-percentage	Umbral bajo del número de ENIs unidas: Umbral alto del número de ENIs unidas  <b>NOTA</b> Este parámetro se descarta. Utilice los otros cuatro parámetros de precalentamiento dinámico de la ENI.	Predeterminado: 0:0

**Paso 4** Haga clic en **OK**.

---Fin

## Referencias

- [kube-apiserver](#)
- [kube-controller-manager](#)
- [kube-scheduler](#)

## 2.6.2 Control de sobrecarga de clúster

### Escenario

Si está habilitado, las solicitudes simultáneas se controlan dinámicamente en función de la presión de recursos de los nodos maestros para mantenerlas disponibles y el clúster.

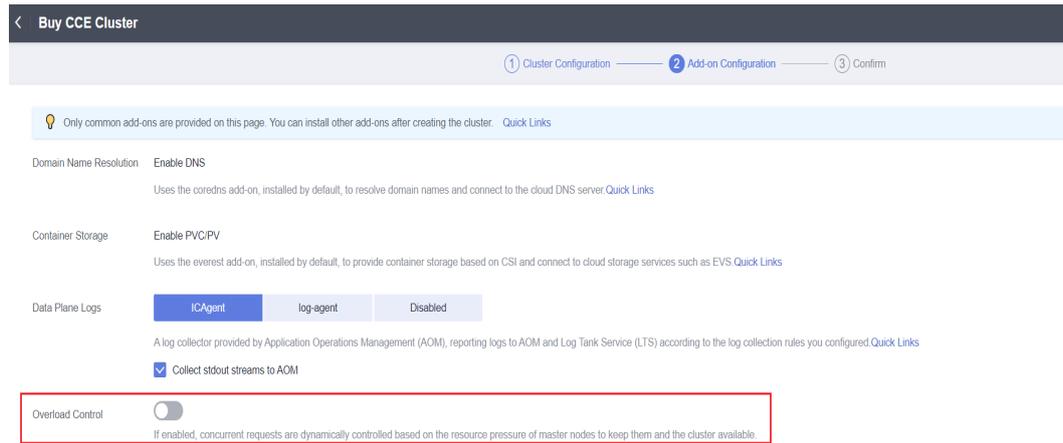
### Restricciones

La versión del clúster es v1.23 o posterior.

### Activación del control de sobrecarga

#### Método 1: Activarlo mediante la creación del clúster

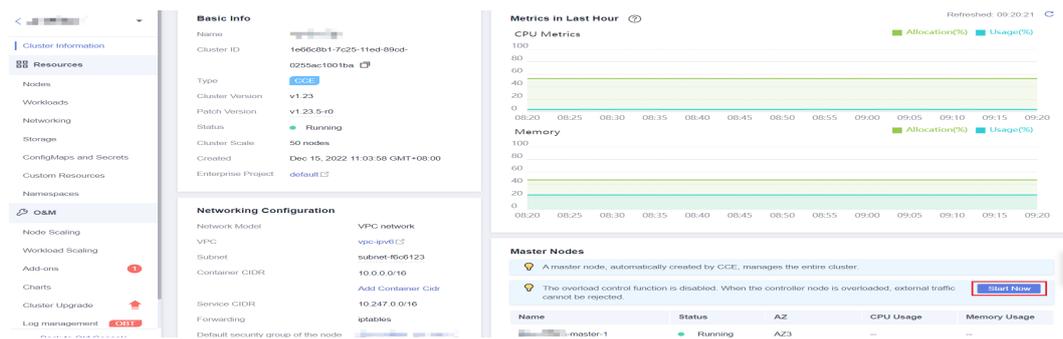
Al crear un clúster de v1.23 o posterior, puede habilitar el control de sobrecarga durante la creación del clúster.



## Método 2: Activarlo en un clúster existente

**Paso 1** Inicie sesión en la consola de CCE y vaya a un clúster existente cuya versión sea v1.23 o posterior.

**Paso 2** En la página de información del clúster, vea la información del nodo principal. Si el control de sobrecarga no está habilitado, se muestra un mensaje. Puede hacer clic en **Enable** para activar la función.



----Fin

## Deshabilitar el control de sobrecarga de clúster

**Paso 1** Inicie sesión en la consola de CCE y vaya a un clúster existente cuya versión sea v1.23 o posterior.

**Paso 2** En la página **Cluster Information**, haga clic en **Manage** en la esquina superior derecha.

**Paso 3** Establezca **support-overload** en **false** bajo **kube-apiserver**.

**Paso 4** Haga clic en **OK**.

----Fin

## 2.6.3 Cambio de escala de clúster

### Escenario

CCE permite cambiar el número de nodos gestionados en un clúster.

## Notas y restricciones

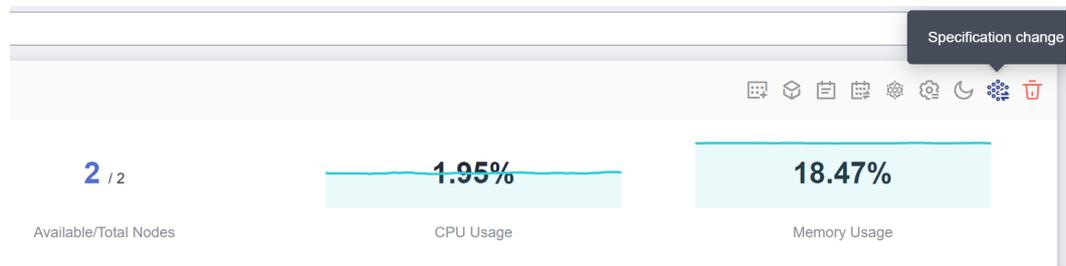
- Esta función es compatible con clústeres de v1.15 y versiones posteriores.
- A partir de la versión 1.15.11, el número de nodos de un clúster se puede cambiar a 2000. El número de nodos en un nodo con un maestro único no se puede cambiar a 1000 o más.
- Actualmente, un clúster solo se puede ampliar a una especificación más grande, pero no se puede reducir.
- Durante el cambio de especificaciones, los nodos principales se apagarán y se encenderán, y el clúster no puede ejecutarse correctamente. Realice el cambio durante las horas fuera de pico.
- El cambio de escala del clúster no afecta a los servicios que se ejecutan en el clúster. Sin embargo, el plano de control (nodos principales) se interrumpirá durante un corto período de tiempo. Se recomienda no realizar ninguna otra operación (como la creación de cargas de trabajo) durante el cambio.
- Los errores de cambio activarán una reversión del clúster al estado normal. Si la reversión falla, envíe un ticket de servicio.

## Procedimiento

**Paso 1** Inicie sesión en la consola de CCE. En el panel de navegación, elija **Clusters**.

**Paso 2** Haga clic en  junto al clúster cuyas especificaciones deben cambiarse.

**Figura 2-15** Cambio de especificaciones

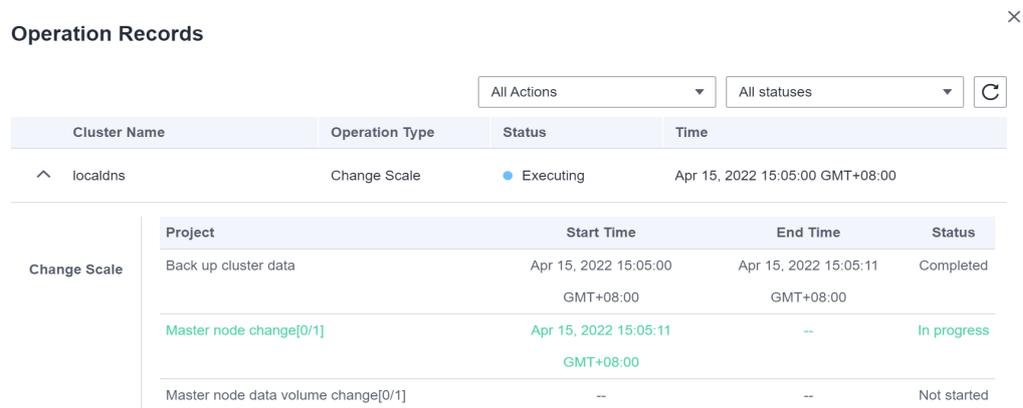


**Paso 3** En la página mostrada, seleccione una nueva variante para **Target Flavor** según sea necesario.

**Paso 4** Haga clic en **OK**.

Puede hacer clic en **Operation Records** en la esquina superior izquierda para ver el historial de cambios del clúster. El estado cambia de **Executing** a **Successful** lo que indica que las especificaciones del clúster se han cambiado correctamente.

**Figura 2-16** Registros de operaciones



Cluster Name	Operation Type	Status	Time
localdns	Change Scale	Executing	Apr 15, 2022 15:05:00 GMT+08:00

Project	Start Time	End Time	Status
Back up cluster data	Apr 15, 2022 15:05:00 GMT+08:00	Apr 15, 2022 15:05:11 GMT+08:00	Completed
Master node change[0/1]	Apr 15, 2022 15:05:11 GMT+08:00	--	In progress
Master node data volume change[0/1]	--	--	Not started

----Fin

## 2.6.4 Eliminación de un clúster

### Escenario

- Puede eliminar directamente los clústeres de pago por uso. Para obtener más información, véase [Eliminación de un clúster](#).
- Los clústeres anuales/mensuales no se pueden eliminar directamente. Debe darse de baja de los clústeres que no han caducado o liberar clústeres que han caducado y no se han renovado. Para obtener más información, véase [Cancelación de la suscripción o liberación de un clúster](#).

### Precauciones

- Cuando se elimina un clúster, los nodos gestionados y los suscritos anualmente/mensualmente se eliminarán del clúster y el sistema se reinstalará. Las contraseñas de inicio de sesión originales de los nodos no serán válidas. Para obtener más información, consulte [Restablecer la contraseña para iniciar sesión en un ECS en la consola de gestión](#).
- La eliminación de un clúster no eliminará los recursos facturados anualmente/mensualmente en el clúster, y su facturación continúa.
- Al eliminar un clúster se eliminarán los nodos del clúster (excepto los nodos aceptados), los discos de datos conectados a los nodos, las cargas de trabajo y los servicios. Los servicios relacionados no se pueden restaurar. Antes de realizar esta operación, asegúrese de que se ha realizado una copia de respaldo o migrado de los datos. Los datos eliminados no pueden restablecerse.

Los recursos que no se crean en CCE no se eliminarán:

- Nodos aceptados (solo se eliminan los nodos creados en CCE);
- Balanceadores de carga de ELB asociados con Services y entradas (solo se eliminan los balanceadores de carga creados automáticamente);
- Recursos de almacenamiento en la nube creados manualmente asociados con PV o recursos de almacenamiento en la nube importados (solo se eliminan los recursos de almacenamiento en la nube creados automáticamente por los PVC)
- Si elimina un clúster que no se está ejecutando (por ejemplo, bloqueado o no disponible), los recursos asociados, como los recursos de almacenamiento y redes, permanecerán.

- Si la versión del clúster es v1.13.10 o anterior, no cambie manualmente el nombre de oyente y el nombre del servidor backend en la consola de ELB. De lo contrario, existirán los recursos residuales cuando elimine el clúster.

## Eliminación de un clúster

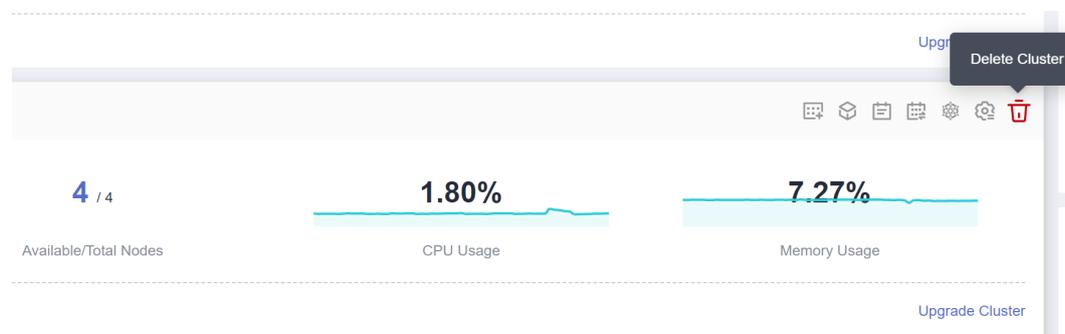
### AVISO

No se puede eliminar un clúster hibernado. Despierte el clúster e inténtelo de nuevo.

**Paso 1** Inicie sesión en la consola de CCE. En el panel de navegación, elija **Clusters**.

**Paso 2** Haga clic en  junto al clúster que desea eliminar.

**Figura 2-17** Eliminación de un clúster



**Paso 3** En el cuadro de diálogo **Delete Cluster** que se muestra, seleccione los recursos que desea liberar.

- Elimine los recursos de almacenamiento en la nube conectados a las cargas de trabajo del clúster.

### **NOTA**

Antes de eliminar los PVC y volúmenes, preste atención a las siguientes reglas:

- Los recursos de almacenamiento subyacentes se eliminan de acuerdo con la política de recuperación que haya definido.
- Si hay un gran número de archivos (más de 1,000) en el bucket de OBS, borre manualmente los archivos y, a continuación, elimine el clúster.
- Eliminar recursos de red, como balanceadores de carga en un clúster. (Solo se pueden eliminar los balanceadores de carga creados automáticamente)
- Eliminar el flujo de log principal de LTS. (Solo se pueden eliminar los flujos de log creados automáticamente)

### **NOTA**

Si no elimina el flujo de log, los logs existentes no se eliminarán, incurriendo en una tarifa. Para obtener más información, consulte [Calculadora de precios](#).

**Paso 4** Haga clic en **Yes** para comenzar a eliminar el clúster.

La operación de eliminación tarda de 1 a 3 minutos en completarse.

----Fin

## Cancelación de la suscripción o liberación de un clúster

### AVISO

- Cuando se cancela la suscripción o se libera un clúster, solo se cancela la suscripción a los recursos asociados con el pedido. Los recursos no asociados se retienen y su facturación continúa.
- Para un clúster anual/mensual, si la retención expira, el clúster se liberará automáticamente. Para los nodos del clúster, si caducan al mismo tiempo, también se liberarán. Si no, CCE no realizará ninguna operación en sus nodos. Los datos del nodo se conservan y la facturación continúa. Preste atención a los clústeres caducados de su cuenta y renuévelos de manera oportuna para evitar la pérdida de datos causada por la reinstalación del nodo.
- Si un pedido incluye recursos de una relación primaria-secundaria, es necesario cancelar la suscripción a los recursos de forma separada.
- Para obtener más información sobre las reglas de cancelación de suscripción, consulte [Cancelación de suscripción permitida](#).
  - Si cancela la suscripción de un recurso que está siendo utilizado, revise la información del recurso y la información de reembolso cuidadosamente. El recurso no se puede restaurar después de cancelar la suscripción. Puede darse de baja de un período de renovación que aún no ha tenido efecto en los recursos renovados en la página [Darse de baja del período de renovación](#).
  - Darse de baja de un recurso asociado con otros recursos facturados anualmente/mensualmente puede afectar el uso normal de esos recursos.

La cancelación de la suscripción de un recurso asociado con otros recursos de pago por uso no afectará al uso normal de esos recursos. Se facturarán normalmente.
  - Si su operación no es una cancelación incondicional de cinco días, se le cobrará la tarifa de gestión y el importe utilizado. Los cupones en efectivo usados y los cupones de descuento no serán reembolsados.

**Paso 1** Inicie sesión en la consola de CCE. En el panel de navegación, elija **Clusters**.

**Paso 2** Haga clic en  junto al clúster de destino.

**Figura 2-18** Cancelar la suscripción de un clúster



**Paso 3** En la página mostrada, seleccione los recursos que desea liberar.

- Elimine los recursos de almacenamiento en la nube conectados a las cargas de trabajo del clúster.

 **NOTA**

Antes de eliminar los PVC y volúmenes, preste atención a las siguientes reglas:

- Los recursos de almacenamiento subyacentes se eliminan de acuerdo con la política de recuperación que haya definido.
- Si hay un gran número de archivos (más de 1,000) en el bucket de OBS, borre manualmente los archivos y, a continuación, elimine el clúster.
- Eliminar recursos de red, como balanceadores de carga en un clúster. (Solo se pueden eliminar los balanceadores de carga creados automáticamente)
- Eliminar el flujo de log principal de LTS. (Solo se pueden eliminar los flujos de log creados automáticamente)

 **NOTA**

Si no elimina el flujo de log, los logs existentes no se eliminarán, incurriendo en una tarifa. Para obtener más información, consulte [Calculadora de precios](#).

**Paso 4** Haga clic en **Yes**. La cancelación de la suscripción o liberación tarda de 1 a 3 minutos en completarse.

----Fin

## 2.6.5 Renewing a Yearly/Monthly-Billed Cluster

You can renew a yearly/monthly-billed cluster.

### Procedure

This section describes how to renew a **yearly/monthly-billed** CCE cluster.

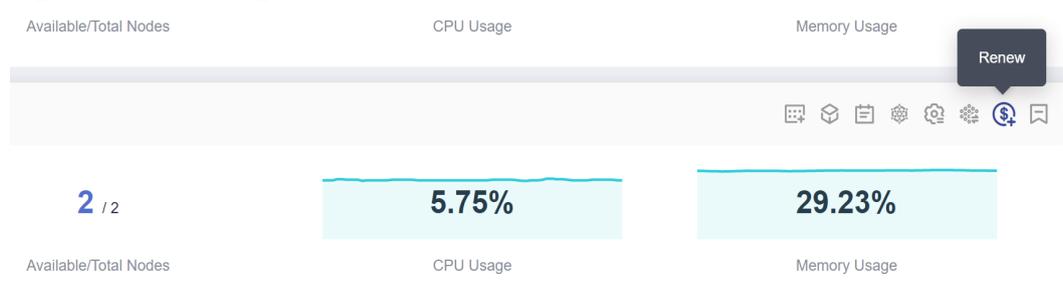
**AVISO**

A yearly/monthly-billed cluster will be deleted if it is not renewed after expiration, and all nodes and the running services in the cluster will be destroyed. CCE strongly recommends that you renew the cluster before it expires or [enable auto renewal](#).

**Paso 1** Log in to the CCE console. In the navigation pane, choose **Clusters**.

**Paso 2** Click  next to the cluster to be renewed.

**Figura 2-19** Renewing a cluster



**Paso 3** On the displayed page, renew the service as prompted.

 **NOTA**

- If the selected resource (highlighted) is associated with other resources, you can decide whether you want to perform the operation on all these resources at the same time.
- If you change resource specifications before the renewal period starts, you can unsubscribe from the resource, but not the renewal period.
- Renewed resources are not eligible of a 5-day unconditional unsubscription.

**Paso 4** Click **Pay**. On the page displayed, review the order amount, select a payment method, and click **Pay**.

**Paso 5** After the payment is complete, you can go back to the **Orders** or **Renewals** page to view and manage your order.

---Fin

## 2.6.6 Hibernación y activación de un clúster (pago por uso)

### Escenario

Si no necesita utilizar un clúster temporalmente, se recomienda hibernar el clúster.

Después de hibernar un clúster, recursos como cargas de trabajo no se pueden crear ni gestionar en el clúster.

Un clúster hibernado puede despertarse rápidamente y usarse normalmente.

### Restricciones

Los clústeres facturados anualmente/mensualmente no se pueden hibernar.

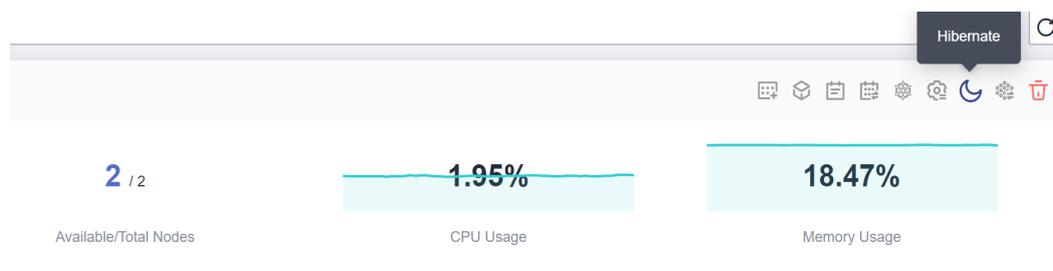
Durante la activación del clúster, es posible que el nodo principal no se inicie debido a la insuficiencia de recursos. Como resultado, el clúster no puede ser despertado. Espere un rato y vuelva a activar el clúster.

### Hibernación de un clúster

**Paso 1** Inicie sesión en la consola de CCE. En el panel de navegación, elija **Clusters**.

**Paso 2** Haga clic en  junto al clúster que se va a hibernar.

**Figura 2-20** Hibernación de un clúster

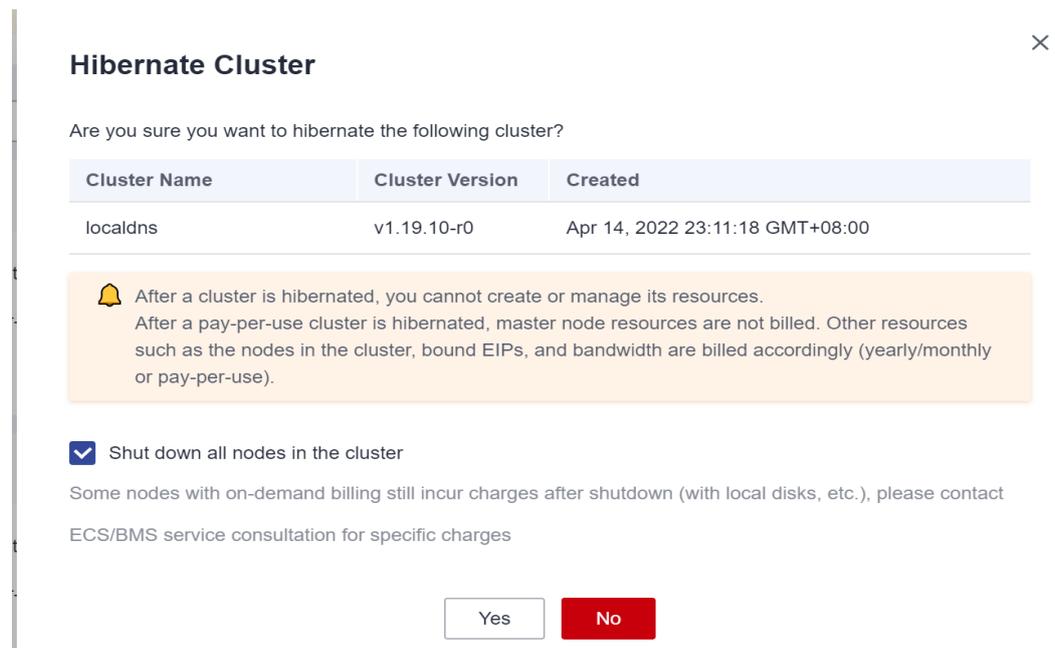


**Paso 3** En el cuadro de diálogo que aparece, compruebe las precauciones y haga clic en **Yes**. Espere hasta que el clúster esté hibernado.

Después de hibernar un clúster, se detendrá la facturación de los recursos del nodo principal. Los recursos, como los nodos de trabajo (ECS), las EIP enlazadas y el ancho de banda, todavía se facturan en función de sus propios modos de facturación. Para cerrar los nodos, seleccione **Stop all nodes in the cluster** en el cuadro de diálogo o vea [Detención de un nodo](#).

La mayoría de los nodos ya no se facturan después de detenerlos, excluidos ciertos tipos de ECS (unos con discos locales conectados, como ECS con uso intensivo de disco y con capacidad ultraalta de E/S). Para obtener más información, consulte [Facturación de ECS](#).

**Figura 2-21** Información de solicitud



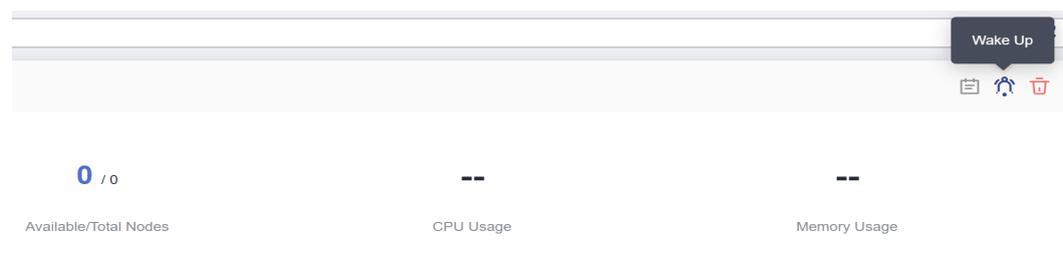
----Fin

## Despierta de un clúster

**Paso 1** Inicie sesión en la consola de CCE. En el panel de navegación, elija **Clusters**.

**Paso 2** Haga clic en  junto al clúster que se va a despertar.

**Figura 2-22** Despierta de un clúster



**Paso 3** Cuando el estado del clúster cambia de **Waking up** a **Running**, se activa el clúster. Se tarda entre 3 y 5 minutos en despertar el clúster.

**NOTA**

Después de activar el clúster, se reanuda la facturación de los recursos en el nodo principal.

----Fin

## 2.6.7 Changing the Billing Mode from Pay-per-Use to Yearly/Monthly

Currently, clusters support **pay-per-use** and **yearly/monthly** billing modes. A pay-per-use cluster can be converted to a yearly/monthly-billed cluster.

### Notes and Constraints

- You cannot change the nodes from pay-per-use to yearly/monthly on the ECS console.
- Only nodes in the default node pool **DefaultPool** can be changed to yearly/monthly billing mode.
- Nodes whose billing mode is changed to yearly/monthly do not support auto scaling.

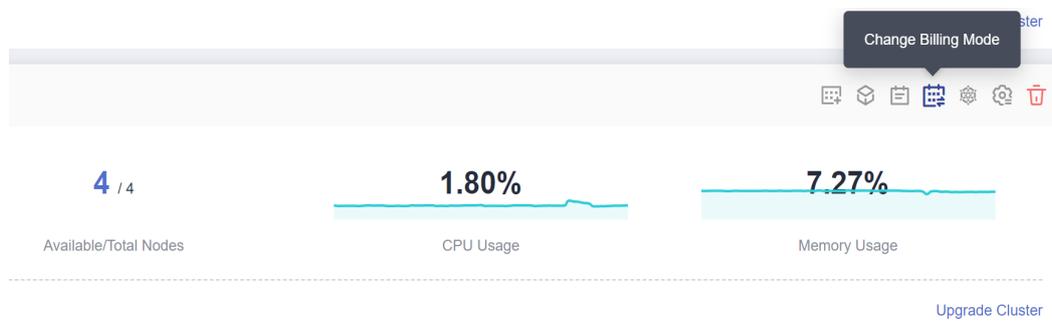
### Changing to Yearly/Monthly Billing

To change the billing mode of the clusters you have purchased from pay-per-use to yearly/monthly, perform the following steps:

**Paso 1** Log in to the CCE console. In the navigation pane, choose **Clusters**.

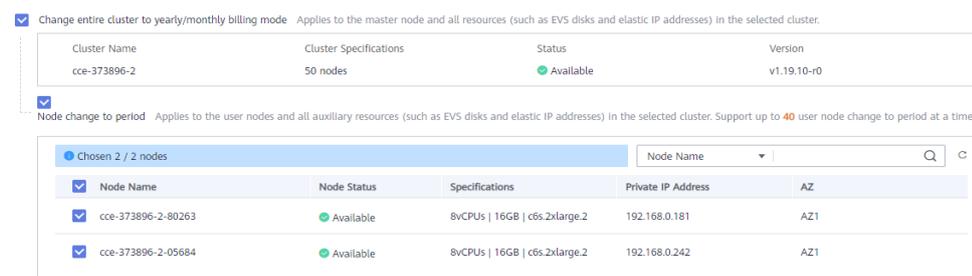
**Paso 2** Click  next to the target cluster.

**Figura 2-23** Changing to the yearly/monthly billing mode



**Paso 3** On the **Change Billing Mode** page, choose the master and worker nodes that will be changed to the yearly/monthly billing mode.

**Figura 2-24** Changing billing mode for master and worker nodes



**Paso 4** Click **OK**. Wait until the order is processed and the payment is complete.

During payment, if a message is displayed indicating that **you do not have the permission to access the resource API**, go back to the previous page and perform the operation again.

----Fin

## 2.6.8 Cambio del grupo de seguridad predeterminado de un nodo

### Escenario

Al crear un clúster, puede personalizar un grupo de seguridad de nodo para gestionar de forma centralizada las políticas de seguridad de red. Para un clúster creado, puede cambiar su grupo de seguridad de nodos predeterminado.

### Restricciones

- No agregue más de 1000 pods al mismo grupo de seguridad. De lo contrario, el rendimiento del grupo de seguridad puede verse afectado. Para obtener más restricciones en los grupos de seguridad, consulte [Restricciones del grupo de seguridad](#).
- No se puede especificar el grupo de seguridad del nodo principal. Tenga cuidado al modificar las reglas del grupo de seguridad del nodo principal. Para obtener más información, consulte [Configuración de reglas de grupo de seguridad de clúster de CCE](#).

### Procedimiento

**Paso 1** Inicie sesión en la consola de CCE. En el panel de navegación, elija **Clusters**.

**Paso 2** Haga clic en el nombre del clúster para acceder a la página de información del clúster.

**Paso 3** En el área **Network Configuration**, haga clic en  junto al **Default security group of the node**.

#### Networking Configuration

Network Model	VPC network
VPC	<a href="#">vpc-cce</a> 
Subnet	<a href="#">subnet-cce</a>
Container CIDR	172.16.0.0/16 <a href="#">Add Container Cidr</a>
Service CIDR	10.247.0.0/16
Forwarding	iptables
Default security group of the node	 <a href="#">4-cce-node-u04rv</a>  

- Paso 4** Seleccione un grupo de seguridad existente, confirme que las reglas del grupo de seguridad cumplen los requisitos del clúster y haga clic en **OK**.

#### AVISO

- Asegúrese de que las reglas de puerto correctas están configuradas para el grupo de seguridad seleccionado. De lo contrario, no se puede crear el nodo. Las reglas de puerto que un grupo de seguridad debe cumplir varían según el tipo de clúster. Para obtener más información, vea la [Configuración de reglas de grupo de seguridad de clúster de CCE](#).
- El nuevo grupo de seguridad solo tiene efecto para los nodos recién creados o gestionados. Para los nodos existentes, modifique las reglas del grupo de seguridad y restablezca los nodos en tiempo real. Se sigue utilizando el grupo de seguridad original. Las reglas de puerto que un grupo de seguridad debe cumplir varían según el tipo de clúster. Para obtener más información, vea la [Configuración de reglas de grupo de seguridad de clúster de CCE](#).

#### Editing the Default Security Group of a Node

Default security group of the node

Sys-default ? C

The default security group of a node needs to allow some ports to pass through. [How to set a security group](#)

如To customize security group rules, you can [add a security group](#). The modified security group applies only to newly created and managed nodes. You need to manually modify the security group rules for existing nodes. [How to Modify](#)

I confirm that the security group has set accurate security group rules to ensure normal communication between nodes.

OK

Cancel

----Fin

# 3 Nodos

---

## 3.1 Descripción del nodo

### 3.1.1 Precauciones para el uso de un nodo

#### Presentación

Un clúster de contenedores consta de un conjunto de máquinas de trabajo, denominadas nodos, que ejecutan aplicaciones en contenedores. Un nodo puede ser una máquina virtual (VM) o una máquina física (PM), dependiendo de sus requisitos de servicio. Los componentes de un nodo incluyen kubelet, tiempo de ejecución del contenedor y kube-proxy.

#### NOTA

Un clúster de Kubernetes consta de nodos maestros y nodos de trabajo. Los nodos descritos en esta sección se refieren a **worker nodes**, los nodos informáticos de un clúster que ejecutan aplicaciones en contenedores.

CCE utiliza Elastic Cloud Servers (ECS) de alto rendimiento o Bare Metal Servers (BMS) como nodos para crear los clústeres de Kubernetes de alta disponibilidad.

#### Especificaciones de nodos compatibles

Las regiones diferentes soportan las variantes de nodo diferentes, y las variantes de nodo pueden cambiarse o agotarse. Se recomienda iniciar sesión en la consola de CCE y comprobar si las variantes de nodo requeridas son compatibles en la página para crear nodos.

#### Sistema de almacenamiento de archivos subyacente de Docker

- En clústeres de v1.15.6 o anterior, el sistema de almacenamiento de archivos subyacente utiliza el formato XFS.
- En clústeres de v1.15.11 o posterior, después de crear o restablecer un nodo, el sistema de almacenamiento de archivos subyacente utiliza el formato ext4.

Para las aplicaciones en contenedores que utilizan el formato XFS, preste atención al impacto del cambio de formato de almacenamiento de archivos subyacente. (La secuencia de archivos en diferentes sistemas de archivos es diferente. Por ejemplo, algunas aplicaciones Java hacen

referencia a un paquete de JAR, pero el directorio contiene varias versiones del paquete de JAR. Si no se especifica la versión, el archivo de sistema determina el paquete real al que se hace referencia.)

Ejecute el comando `docker info | grep "Backing Filesystem"` para comprobar el formato del archivo de almacenamiento subyacente de Docker utilizado por el nodo actual.

## Usuario y grupo de usuarios de Paas

Cuando se crea un nodo en un clúster de CCE, se crea un usuario o grupo de usuarios paas en el nodo de forma predeterminada. Los componentes de CCE y los complementos de CCE en un nodo se ejecutan como un usuario no root (usuario/grupo de usuarios de paas) para minimizar el permiso de ejecución. Si se modifica el usuario o grupo de usuarios de paas, es posible que los componentes y pods de CCE no se ejecuten correctamente.

### AVISO

El funcionamiento normal de los componentes de CCE depende del usuario o grupo de usuarios paas. Preste atención a los siguientes requisitos:

- No modifique el permiso de directorio y el permiso de directorio contenedor en un nodo.
- No cambie el GID y el UID del usuario o grupo de usuarios de paas.
- No utilice directamente el usuario o grupo de usuarios de paas para establecer el usuario y el grupo al que pertenece el archivo de servicio.

## Ciclo de vida del nodo

Un ciclo de vida indica los estados de nodo registrados desde el momento en que se crea el nodo hasta el momento en que se elimina o se libera el nodo.

**Tabla 3-1** Estado de nodo

Estado	Atributo de estado	Descripción
Running	Estado estable	El nodo se ejecuta correctamente y está conectado al clúster. Los nodos en este estado pueden proporcionar servicios.
Unavailable	Estado estable	El nodo no se está ejecutando correctamente. Las instancias en este estado ya no proporcionan servicios. En este caso, realice las operaciones de <a href="#">Restablecimiento de un nodo</a> .
Creating	Estado intermedio	El nodo se ha creado pero no se está ejecutando.
Installing	Estado intermedio	El software de Kubernetes se está instalando en el nodo.

Estado	Atributo de estado	Descripción
Deleting	Estado intermedio	El nodo se está eliminando. Si este estado permanece durante mucho tiempo, se produce una excepción.
Stopped	Estado estable	El nodo se detiene correctamente. Un nodo en este estado no puede proporcionar servicios. Puede iniciar el nodo en la consola de ECS.
Error	Estado estable	El nodo es anormal. Las instancias en este estado ya no proporcionan servicios. En este caso, realice las operaciones de <b>Restablecimiento de un nodo</b> .

## 3.1.2 Descripción del motor de contenedores

### Introducción a los motores de contenedores

Los motores de contenedores, uno de los componentes más importantes de Kubernetes, gestionan el ciclo de vida de las imágenes y los contenedores. El kubelet interactúa con un tiempo de ejecución contenedor con la Container Runtime Interface (CRI).

CCE soporta containerd y Docker. **containerd se recomienda por sus trazas más cortas, menos componentes, mayor estabilidad y menos consumo de recursos de nodo.**

Kubernetes ha eliminado dockershim de v1.24 y no es compatible con Docker de forma predeterminada. Para obtener más información, consulta [Kubernetes está pasando de Dockershim: Compromisos y próximos pasos](#). Para garantizar la experiencia del usuario, CCE seguirá soportando Docker en la v1.25, pero solo hasta la v1.27. Es necesario migrar los nodos de Docker a containerd antes de eso. Para obtener más información, véase [Migración de nodos de Docker a containerd](#).

### Asignación entre los sistemas operativos de nodos y los motores de contenedores

Tabla 3-2 Sistemas operativos de nodo y motores de contenedor en clústeres de CCE

SO	Versión del kernel	Motor de contenedores	Container Storage Rootfs	Container Runtime
CentOS 7.x	3.x	Docker Los clústeres de v1.23 y posteriores admiten containerd.	Los clústeres de v1.19.16 y anteriores usan Device Mapper. Los clústeres de v1.19.16 y posteriores usan OverlayFS.	runC

SO	Versión del kernel	Motor de contenedores	Container Storage Rootfs	Container Runtime
EulerOS 2.3	3.x	Docker	Device Mapper	runC
EulerOS 2.5	3.x	Docker	Device Mapper	runC
EulerOS 2.9	4.x	Docker Los clústeres de v1.23 y posteriores admiten containerd.	OverlayFS	runC
Ubuntu 18.04	4.x	Docker Los clústeres de v1.23 y posteriores admiten containerd.	OverlayFS	runC
Huawei Cloud EulerOS 1.1	3.x	Docker containerd	OverlayFS	runC
Huawei Cloud EulerOS 2.0	5.x	Docker containerd	OverlayFS	runC

**Tabla 3-3** Sistemas operativos de nodo y motores de contenedor en clústeres de Turbo de CCE

Tipo de nodo	SO	Versión del kernel	Motor de contenedores	Rootfs de almacenamiento de contenedor	Tiempo de ejecución de contenedores
Elastic Cloud Server (VM)	CentOS 7.6	3.x	Docker containerd	OverlayFS	runC
	Ubuntu 18.04	4.x			
	EulerOS 2.9	4.x			
	Huawei Cloud EulerOS 1.1	3.x			
	Huawei Cloud EulerOS 2.0	5.x			

Tipo de nodo	SO	Versión del kernel	Motor de contenedores	Rootfs de almacenamiento de contenedor	Tiempo de ejecución de contenedores
Elastic Cloud Server (physical machine)	EulerOS 2.9	4.x	containerd	Device Mapper	Kata

**Tabla 3-4** Sistemas operativos de nodo y motores de contenedor en CCE de Kunpeng

SO	Versión del kernel	Motor de contenedores	Rootfs de almacenamiento de contenedor	Tiempo de ejecución de contenedores
EulerOS 2.8	4.x	Docker	OverlayFS	runC

## Comandos comunes de containerd y Docker

containerd no soporta Docker API y Docker CLI, pero puede ejecutar comandos crictl para implementar funciones similares.

**Tabla 3-5** Comandos relacionados con la imagen

N.º	Comando de Docker	Comando de containerd	Notas
1	docker images [Option] [Image name[:Tag]]	crictl images [Option] [Image name[:Tag]]	Listar imágenes locales.
2	docker pull [Option] <i>Image name[:Tag@DIGEST]</i>	crictl pull [Option] <i>Image name[:Tag@DIGEST]</i>	Extraer las imágenes.
3	docker push	Ninguno	Empujar las imágenes.
4	docker rmi [Option] <i>Image...</i>	crictl rmi [Option] <i>Image ID...</i>	Eliminar una imagen local.
5	docker inspect <i>Image ID</i>	crictl inspecti <i>Image ID</i>	Comprobar las imágenes.

**Tabla 3-6** Comandos relacionados con el contenedor

N.º	Comando de Docker	Comando de containerd	Notas
1	docker ps [Option]	crictl ps [Option]	Listar los contenedores.
2	docker create [Option]	crictl create [Option]	Crear un contenedor.
3	docker start [Option] <i>Container ID...</i>	crictl start [Option] <i>Container ID...</i>	Iniciar un contenedor.
4	docker stop [Option] <i>Container ID...</i>	crictl stop [Option] <i>Container ID...</i>	Detener un contenedor.
5	docker rm [Option] <i>Container ID...</i>	crictl rm [Option] <i>Container ID...</i>	Eliminar un contenedor.
6	docker attach [Option] <i>Container ID</i>	crictl attach [Option] <i>Container ID</i>	Conectarse a un contenedor.
7	docker exec [Option] <i>Container ID Startup command [Parameter...]</i>	crictl exec [Option] <i>Container ID Startup command [Parameter...]</i>	Acceder al contenedor.
8	docker inspect [Option] <i>Container name ID...</i>	crictl inspect [Option] <i>Container ID...</i>	Consultar detalles del contenedor.
9	docker logs [Option] <i>Container ID</i>	crictl logs [Option] <i>Container ID</i>	Ver los logs de contenedor.
10	docker stats [Option] <i>Container ID...</i>	crictl stats [Option] <i>Container ID</i>	Comprobar el uso de recursos del contenedor.
11	docker update [Option] <i>Container ID...</i>	crictl update [Option] <i>Container ID...</i>	Actualizar los límites de recursos de contenedor.

**Tabla 3-7** Comandos relacionados con pod

N.º	Comando de Docker	Comando de containerd	Notas
1	Ninguno	crictl pods [Option]	Listar los pods.
2	Ninguno	crictl inspectp [Option] <i>Pod ID...</i>	Ver detalles del pod.
3	Ninguno	crictl start [Option] <i>Pod ID...</i>	Iniciar un pod.
4	Ninguno	crictl runp [Option] <i>Pod ID...</i>	Ejecutar un pod.

N.º	Comando de Docker	Comando de containerd	Notas
5	Ninguno	crictl stopp [Option] Pod ID...	Detener un pod.
6	Ninguno	crictl rmp [Option] Pod ID...	Borra un pod.

 **NOTA**

Los contenedores creados e iniciados por containerd son eliminados inmediatamente por kubelet. containerd no admite la suspensión, reanudación, reinicio, cambio de nombre y espera de contenedores ni la creación, importación, exportación, comparación, inserción, búsqueda y etiquetado de imágenes de Docker. containerd no admite la copia de archivos. Puede iniciar sesión en el repositorio de imágenes modificando el archivo de configuración de containerd.

### Diferencias en el rastreo

- Docker (Kubernetes 1.23 y las versiones anteriores):  
 kubelet --> docker shim (en el proceso de kubelet) --> docker --> containerd
- Docker (solución comunitaria para Kubernetes v1.24 o posterior):  
 kubelet --> cri-dockerd (kubelet utiliza CRI para conectarse a cri-dockerd) --> docker--> containerd
- containerd:  
 kubelet --> cri plugin (en el proceso de containerd) --> containerd

Aunque Docker ha agregado funciones como clúster de enjambre, construcción de Docker y API de Docker, también introduce errores. En comparación con containerd, Docker tiene una capa más de llamadas. **Por lo tanto, containerd ahorra más recursos y es más seguro.**

### Descripción de la versión del motor de contenedores

- Docker
  - EulerOS/CentOS: docker-engine 18.9.0, una versión de Docker personalizada para CCE. Las vulnerabilidades de seguridad se corregirán de manera oportuna.
  - Ubuntu: docker-ce 18.9.9 (versión comunitaria). Se recomienda utilizar el motor de containerd para los nodos de Ubuntu.

 **NOTA**

El docker-ce de código abierto de Ubuntu puede activar errores cuando se realizan operaciones de ejecución simultánea (por ejemplo, se configuran múltiples sondeos de ejecución). Se recomienda utilizar sondas de HTTP/TCP.

- containerd: 1.4.1

## 3.1.3 Descripción del nodo del SO

### Asignaciones entre versiones de clúster y versiones de sistema operativo

En la siguiente tabla se enumeran las asignaciones entre las versiones de clúster publicadas y las versiones de SO.

**Tabla 3-8** Asignaciones entre las versiones del SO del nodo de VM y las versiones del clúster

Versión del SO	Versión del clúster	Último Kernel
Huawei Cloud EulerOS 2.0	v1.25	5.10.0-60.18.0.50.r865_35.hce2.x86_64
	v1.23	5.10.0-60.18.0.50.r865_35.hce2.x86_64
Huawei Cloud EulerOS 2.0 (ARM)	v1.25	5.10.0-60.18.0.50.r865_35.hce2.aarch64
	v1.23	5.10.0-60.18.0.50.r865_35.hce2.aarch64
Ubuntu 22.04	1.25	5.15.0-60-generic
	1.23	5.15.0-60-generic
CentOS Linux release 7.6	v1.25	3.10.0-1160.66.1.el7.x86_64
	v1.23	3.10.0-1160.66.1.el7.x86_64
	v1.21	3.10.0-1160.66.1.el7.x86_64
	v1.19.16	3.10.0-1160.66.1.el7.x86_64
	v1.19.10	3.10.0-1160.25.1.el7.x86_64
	v1.19.8	3.10.0-1160.15.2.el7.x86_64
	v1.17.17 (Fin del mantenimiento)	3.10.0-1160.15.2.el7.x86_64
	v1.17.9 (Fin del mantenimiento)	3.10.0-1062.12.1.el7.x86_64
	v1.15.11 (Fin de mantenimiento)	3.10.0-1062.12.1.el7.x86_64
	v1.15.6-r1 (Fin de mantenimiento)	3.10.0-1062.1.1.el7.x86_64
	v1.13.10-r1 (Fin de mantenimiento)	3.10.0-957.21.3.el7.x86_64
v1.13.7-r0 (Fin del mantenimiento)	3.10.0-957.21.3.el7.x86_64	
EulerOS release 2.9	v1.25	4.18.0-147.5.1.6.h841.eulerosv2r9.x86_64
	v1.23	4.18.0-147.5.1.6.h841.eulerosv2r9.x86_64
	v1.21	4.18.0-147.5.1.6.h841.eulerosv2r9.x86_64
	v1.19	4.18.0-147.5.1.6.h841.eulerosv2r9.x86_64

Versión del SO	Versión del clúster	Último Kernel
EulerOS release 2.9 (ARM)	v1.25	4.19.90-vhulk2103.1.0.h848.eulerosv2r9.aarch64
	v1.23	4.19.90-vhulk2103.1.0.h848.eulerosv2r9.aarch64
	v1.21	4.19.90-vhulk2103.1.0.h848.eulerosv2r9.aarch64
	v1.19	4.19.90-vhulk2103.1.0.h848.eulerosv2r9.aarch64
EulerOS release 2.10	v1.25	4.18.0-147.5.2.10.h933.eulerosv2r10.x86_64
	v1.23	4.18.0-147.5.2.10.h933.eulerosv2r10.x86_64
EulerOS release 2.10 (ARM)	v1.25	4.19.90-vhulk2204.1.0.h1160.eulerosv2r10.aarch64
	v1.23	4.19.90-vhulk2204.1.0.h1160.eulerosv2r10.aarch64
EulerOS release 2.8 (ARM)	v1.25	4.19.36-vhulk1907.1.0.h1350.eulerosv2r8.aarch64
	v1.23	4.19.36-vhulk1907.1.0.h1350.eulerosv2r8.aarch64
	v1.21	4.19.36-vhulk1907.1.0.h1350.eulerosv2r8.aarch64
	v1.19.16	4.19.36-vhulk1907.1.0.h1350.eulerosv2r8.aarch64
	v1.19.10	4.19.36-vhulk1907.1.0.h962.eulerosv2r8.aarch64
	v1.17.17 (End of maintenance)	4.19.36-vhulk1907.1.0.h962.eulerosv2r8.aarch64
	v1.15.11 (End of maintenance)	4.19.36-vhulk1907.1.0.h702.eulerosv2r8.aarch64
EulerOS release 2.5	v1.25	3.10.0-862.14.1.5.h687.eulerosv2r7.x86_64

Versión del SO	Versión del clúster	Último Kernel
	v1.23	3.10.0-862.14.1.5.h687.eulerosv2r7.x86_64
	v1.21	3.10.0-862.14.1.5.h687.eulerosv2r7.x86_64
	v1.19.16	3.10.0-862.14.1.5.h687.eulerosv2r7.x86_64
	v1.19.10	3.10.0-862.14.1.5.h520.eulerosv2r7.x86_64
	v1.19.8	3.10.0-862.14.1.5.h520.eulerosv2r7.x86_64
	v1.17.17 (Fin del mantenimiento)	3.10.0-862.14.1.5.h470.eulerosv2r7.x86_64
	v1.17.9 (Fin del mantenimiento)	3.10.0-862.14.1.5.h428.eulerosv2r7.x86_64
	v1.15.11 (Fin de mantenimiento)	3.10.0-862.14.1.5.h428.eulerosv2r7.x86_64
	v1.15.6-r1 (Fin de mantenimiento)	3.10.0-862.14.1.5.h328.eulerosv2r7.x86_64
	v1.13.10-r1 (Fin de mantenimiento)	3.10.0-862.14.1.2.h249.eulerosv2r7.x86_64
	v1.13.7-r0 (Fin del mantenimiento)	3.10.0-862.14.1.0.h197.eulerosv2r7.x86_64
Ubuntu 18.04 server 64-bit (Fin del mantenimiento)	v1.25	4.15.0-171-generic
	v1.23	4.15.0-171-generic
	v1.21	4.15.0-171-generic
	v1.19.16	4.15.0-171-generic
	v1.19.8	4.15.0-136-generic
	v1.17.17 (Fin del mantenimiento)	4.15.0-136-generic
Huawei Cloud EulerOS 1.1	v1.25	3.10.0-1160.66.1.hce1c.x86_64
	v1.23	3.10.0-1160.66.1.hce1c.x86_64
	v1.21	3.10.0-1160.66.1.hce1c.x86_64

**Tabla 3-9** Asignaciones entre las versiones del SO del nodo de BMS y las versiones del clúster

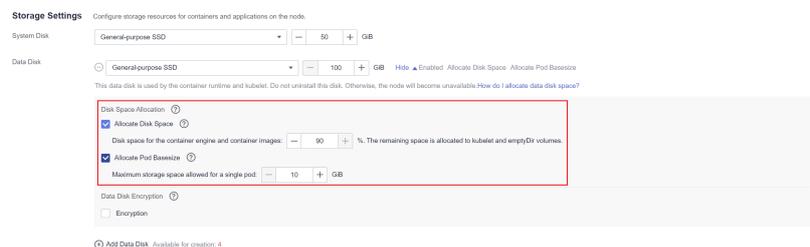
Versión del SO	Versión del clúster	Información del Kernel
EulerOS release 2.10 (servidor elástico de metal desnudo)	v1.25 v1.23 v1.21 v1.19.16	4.18.0-147.5.2.15.h1109.eulerosv2r10.x86_64
EulerOS release 2.9 (servidor de metal desnudo)	v1.25 v1.23 v1.21 v1.19	4.18.0-147.5.1.6.h841.eulerosv2r9.x86_64  Para obtener más información sobre las especificaciones del servidor, consulte la <a href="#">Familia de instancias</a> .
EulerOS versión 2.3 (servidor de metal desnudo, fin de mantenimiento)	v1.15.11 o posterior	3.10.0-514.41.4.28.h62.x86_64  Para obtener más información sobre las especificaciones del servidor, consulte la <a href="#">Familia de instancias</a> .

### 3.1.4 Asignación de espacio en disco de datos

Esta sección describe cómo asignar espacio en disco de datos a los nodos para que pueda configurar el espacio en disco de datos en consecuencia.

#### Asignación de espacio en disco de datos

Al crear un nodo, debe configurar un disco de datos para el nodo y asegurarse de que la capacidad del disco de datos es mayor o igual a 100 GB. Puede hacer clic en **Expand** para personalizar la asignación de espacio en disco de datos del nodo.



- **Asignar espacio en disco:**

CCE divides the data disk space for container engines and pods. The container engine space stores the **Docker/containerd** working directories, container images, and image metadata. The other is reserved for kubelet and emptyDir volumes. The available container engine space affects image pulls and container startup and running.

- Motor de contenedores y espacio de imagen contenedor (90% por defecto): almacena los directorios de trabajo de contenedor en tiempo de ejecución, datos de imagen de contenedor y metadatos de imagen.

- kubelet y espacio de emptyDir (10% por defecto): almacena archivos de configuración de pod, secretos y almacenamiento montado como volúmenes de emptyDir.
- **Asignar tamaño de la base del pod:** indica el tamaño de la base de un contenedor. Puede establecer un límite superior para el espacio en disco ocupado por cada pod de carga de trabajo (incluido el espacio ocupado por las imágenes de contenedor). Esta configuración impide que los pods ocupen todo el espacio disponible en disco, lo que puede provocar excepciones de servicio. Se recomienda que el valor sea inferior o igual al 80% del espacio del motor del contenedor. Este parámetro está relacionado con el sistema operativo del nodo y los rootfs de almacenamiento de contenedor y no se admite en algunos escenarios.

## Asignación de espacio en disco

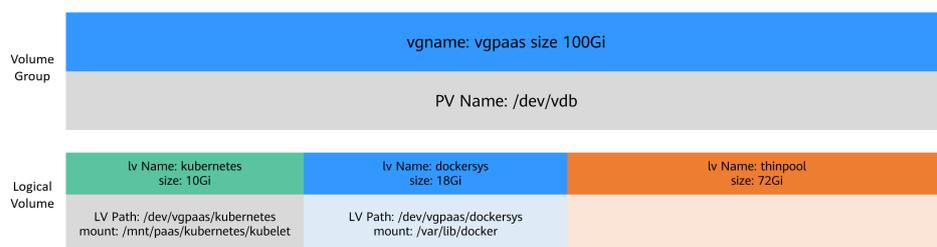
Un disco de datos, 100 GB por ejemplo, se puede dividir de la siguiente manera (dependiendo de los Rootfs de almacenamiento contenedor): Para obtener más información sobre los Rootfs de almacenamiento de contenedor correspondientes a diferentes sistemas operativos, consulte la sección [Asignación entre SO y rootfs de almacenamiento de contenedores](#).

- **Rootfs (Device Mapper)**

De forma predeterminada, el motor de contenedor y el espacio de imagen, que ocupan el 90% del disco de datos, se pueden dividir en las dos partes siguientes:

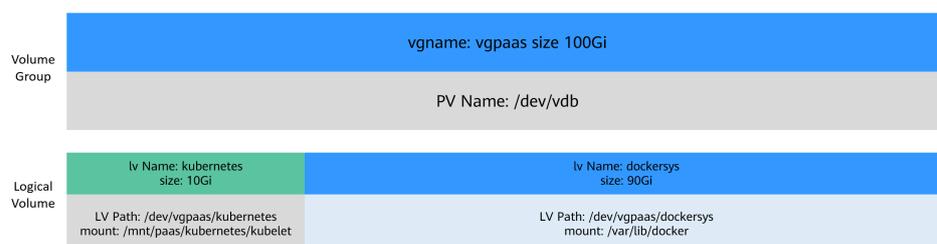
- El directorio **/var/lib/docker** se utiliza como el directorio de trabajo de Docker y ocupa el 20% del motor de contenedor y el espacio de imagen de contenedor por defecto. (Tamaño del espacio del directorio **/var/lib/docker** = **Espacio en disco de datos x 90% x 20%**)
- El thin pool se utiliza para almacenar datos de imagen de contenedor, metadatos de imagen y datos de contenedor, y ocupa el 80% del motor de contenedor y el espacio de imagen de contenedor de forma predeterminada. (Espacio de thin pool = **Espacio de disco de datos x 90% x 80%**)

El thin pool está montado dinámicamente. Puede verlo ejecutando el comando **lsblk** en un nodo, pero no el comando **df -h**.



- **Rootfs (OverlayFS)**

No hay thin pool separado. Todo el motor de contenedor y el espacio de imagen de contenedor (90% del disco de datos por defecto) están en el directorio **/var/lib/docker**.



## Asignación de tamaño de la base para pods

El espacio de contenedor de pod personalizado (tamaño de la base) está relacionado con el SO del nodo y el almacenamiento de contenedor Rootfs. Para obtener más información acerca de los Rootfs de almacenamiento de contenedor, consulte [Asignación entre SO y rootfs de almacenamiento de contenedores](#).

- Device Mapper admite el tamaño de la base de pod personalizado. El valor predeterminado es 10 GB.
- En el modo de OverlayFS, el espacio de contenedor del pod no está limitado de forma predeterminada.

### NOTA

En el caso de usar Docker en los nodos de EulerOS 2.9, el **basesize** no tendrá efecto si **CAP\_SYS\_RESOURCE** o **privileged** están configurados para un contenedor.

Al configurar **basesize**, tenga en cuenta el número máximo de pods en un nodo. El espacio de contenedor del motor debe ser mayor que el espacio total en disco utilizado por contenedores. Fórmula: **el espacio contenedor del motor y el espacio de imagen contenedor (90% por defecto) > Número de contenedores x tamaño de la base**. De lo contrario, el espacio de motor de contenedor asignado al nodo puede ser insuficiente y el contenedor no puede iniciarse.

Max. Pods

Maximum number of pods (including default system pods) that can run properly on a node. This configuration prevents the node from being overloaded by pods. [Learn more](#)



En el caso de los nodos que admiten **basesize**, cuando se utiliza Device Mapper, aunque puede limitar el tamaño del directorio **/home** de un solo contenedor a 10 GB de forma predeterminada, todos los contenedores del nodo siguen compartiendo el grupo delgado del nodo para el almacenamiento. No están completamente aislados. Cuando la suma del espacio de thin pool utilizado por ciertos recipientes alcanza el límite superior, otros recipientes no pueden funcionar correctamente.

Además, después de eliminar un archivo en el directorio **/home** del contenedor, el espacio de thin pool ocupado por el archivo no se libera inmediatamente. Por lo tanto, incluso si **basesize** se establece en 10 GB, el espacio de thin pool ocupado por los archivos sigue aumentando hasta 10 GB cuando se crean archivos en el contenedor. El espacio liberado después de la eliminación del archivo se reutilizará, pero después de un tiempo. Si **el número de contenedores en el nodo multiplicado por tamaño básico** es mayor que el tamaño del espacio de thin pool del nodo, existe la posibilidad de que se haya agotado el espacio de thin pool.

## Asignación entre SO y rootfs de almacenamiento de contenedores

**Tabla 3-10** Sistemas operativos de nodo y motores de contenedor en clústeres de CCE

SO	Rootfs de almacenamiento de contenedor	Tamaño de la base personalizado
CentOS 7.x	Los clústeres de v1.19.16 y anteriores usan Device Mapper. Los clústeres de v1.19.16 y posteriores usan OverlayFS.	Se admite cuando Rootfs se establece en Device Mapper y el motor de contenedor es Docker. El valor predeterminado es 10G. No se admite cuando Rootfs se establece en OverlayFS.
EulerOS 2.3	Device Mapper	Se admite solo cuando el motor de contenedor es Docker. El valor predeterminado es 10G.
EulerOS 2.5	Device Mapper	Se admite solo cuando el motor de contenedor es Docker. El valor predeterminado es 10G.
EulerOS 2.8	Los clústeres de v1.19.16-r2 y anteriores usan Device Mapper. Los clústeres de v1.19.16-r2 y posteriores usan OverlayFS.	Se admite cuando Rootfs se establece en Device Mapper y el motor de contenedor es Docker. El valor predeterminado es 10G. No se admite cuando Rootfs se establece en OverlayFS.
EulerOS 2.9	OverlayFS	Solo admite los clústeres de v1.19.16, v1.21.3, v1.23.3 y posteriores. El tamaño de la base del contenedor no está limitado por defecto. No se admite cuando las versiones del clúster son anteriores a v1.19.16, v1.21.3 y v1.23.3.
EulerOS 2.10	OverlayFS	Se admite solo cuando el motor de contenedor es Docker. El tamaño de la base del contenedor no está limitado por defecto.
Ubuntu 18.04	OverlayFS	No se admite.
Huawei Cloud EulerOS 1.1	OverlayFS	No se admite.
Huawei Cloud EulerOS 2.0	OverlayFS	Se admite solo cuando el motor de contenedor es Docker. El tamaño de la base del contenedor no está limitado por defecto.

**Tabla 3-11** Sistemas operativos de nodo y motores de contenedor en clústeres de Turbo de CCE

SO	Rootfs de almacenamiento de contenedor	Tamaño de la base personalizado
CentOS 7.x	OverlayFS	No se admite.
Ubuntu 18.04	OverlayFS	No se admite.
EulerOS 2.9	ECS VMs use OverlayFS. ECS PMs use Device Mapper.	Solo se admite cuando Rootfs se establece en OverlayFS y el motor de contenedor es Docker. El tamaño de la base del contenedor no está limitado por defecto.  Se admite cuando Rootfs se establece en Device Mapper y el motor de contenedor es Docker. El valor predeterminado es 10G.
Huawei Cloud EulerOS 1.1	OverlayFS	No se admite.
Huawei Cloud EulerOS 2.0	OverlayFS	Se admite solo cuando el motor de contenedor es Docker. El tamaño de la base del contenedor no está limitado por defecto.

## Políticas de recolección de basura para imágenes de contenedores

Cuando el espacio del motor de contenedor es insuficiente, se activa la recolección de basura de imagen.

La política para la recolección de imágenes de basura tiene en cuenta dos factores: **HighThresholdPercent** y **LowThresholdPercent**. El uso del disco por encima del umbral alto (predeterminado: 85%) activará la recolección de basura. La recolección de basura eliminará las imágenes utilizadas menos recientemente hasta que se cumpla el umbral bajo (por defecto: 80%).

## Configuración recomendada para el espacio del motor del contenedor

- El espacio de contenedor del motor debe ser mayor que el espacio total en disco utilizado por contenedores. Fórmula: **Espacio del motor del contenedor > Número de contenedores x tamaño de la base**
- Se recomienda crear y eliminar archivos de servicios en contenedores en volúmenes de almacenamiento locales (como volúmenes de emptyDir y de hostPath) o directorios de almacenamiento en la nube montados en contenedores. De esta manera, el espacio de thin pool no está ocupado. Los volúmenes de emptyDir ocupan el espacio de kubelet. Por lo tanto, planifique correctamente el tamaño del espacio de kubelet.
- Puede desplegar servicios en nodos que utilizan OverlayFS (para obtener más información, consulte [Asignación entre SO y rootfs de almacenamiento de](#)

**contenedores**) para que el espacio en disco ocupado por los archivos creados o eliminados de contenedores se pueda liberar inmediatamente.

## Problemas comunes

[¿Cómo puedo ampliar la capacidad de almacenamiento de un contenedor?](#)

[Ampliación de la capacidad del disco de un nodo en un clúster de CCE](#)

### 3.1.5 Descripción de los recursos de nodos reservados

Algunos de los recursos del nodo necesitan ejecutar algunos componentes y recursos necesarios del sistema de Kubernetes para que el nodo forme parte de su clúster. Por lo tanto, el número total de recursos de nodo y el número de recursos de nodo asignables en Kubernetes son diferentes. Cuanto más grandes sean las especificaciones del nodo, más se desplegará los contenedores en el nodo. Por lo tanto, es necesario reservar más recursos de nodo para ejecutar los componentes de Kubernetes.

Para garantizar la estabilidad del nodo, se reservará una cierta cantidad de recursos de nodo de CCE para los componentes de Kubernetes (como kubelet, kube-proxy y docker) en función de las especificaciones del nodo.

CCE calcula los recursos que se pueden asignar a los nodos de usuario de la siguiente manera:

**Recursos asignables = Importe total - Importe reservado - Umbral de desalojo**

El umbral de desahucio de memoria se fija en 100 MB.

Cuando aumenta la memoria consumida por todos los pods de un nodo, pueden producirse los siguientes comportamientos:

1. Cuando la memoria disponible del nodo es menor que el umbral de desalojo, kubelet se activa para desalojar el pod. Para obtener más información sobre el umbral de desalojo en Kubernetes, consulte [Desalojo de presión de nodo](#).
2. Si un nodo activa un evento de insuficiencia de memoria del sistema operativo (OOM) antes de que el kubelet recupere la memoria, el sistema termina el contenedor. Sin embargo, a diferencia del desalojo del pod, kubelet reinicia el contenedor basado en el RestartPolicy del pod.

### Reglas para la reserva de memoria de nodo (v1)

Para grupos de **v1.21.4-r0**, **v1.23.3-r0** o posteriores, el modelo de reserva de memoria de nodo se optimiza a V2. Para obtener más información, véase [Reglas para reservar memoria de nodo \(v2\)](#).

Puede utilizar la siguiente fórmula para calcular cuánta memoria debe reservar para ejecutar contenedores en un nodo:

Cantidad total reservada = memoria reservada para componentes del sistema + memoria reservada para kubelet para gestionar pods

**Tabla 3-12** Reglas de reserva para componentes del sistema

Memoria total (TM)	Memoria reservada para componentes del sistema
$TM \leq 8 \text{ GB}$	0 MB

Memoria total (TM)	Memoria reservada para componentes del sistema
$8 \text{ GB} < TM \leq 16 \text{ GB}$	$[(TM - 8 \text{ GB}) \times 1024 \times 10\%] \text{ MB}$
$16 \text{ GB} < TM \leq 128 \text{ GB}$	$[8 \text{ GB} \times 1024 \times 10\% + (TM - 16 \text{ GB}) \times 1024 \times 6\%] \text{ MB}$
$TM > 128 \text{ GB}$	$(8 \text{ GB} \times 1024 \times 10\% + 112 \text{ GB} \times 1024 \times 6\% + (TM - 128 \text{ GB}) \times 1024 \times 2\%) \text{ MB}$

**Tabla 3-13** Reglas de reserva para kubelet

Memoria total (TM)	Número de pods	Memoria reservada para kubelet
$TM \leq 2 \text{ GB}$	-	$TM \times 25\%$
$TM > 2 \text{ GB}$	$0 < \text{Máx. pods en un nodo} \leq 16$	700 MB
	$16 < \text{Máx. pods en un nodo} \leq 32$	$[700 + (\text{máx. pods en un nodo} - 16) \times 18.75] \text{ MB}$
	$32 < \text{Máx. pods en un nodo} \leq 64$	$[1024 + (\text{máx. pods en un nodo} - 32) \times 6.25] \text{ MB}$
	$64 < \text{Máx. pods en un nodo} \leq 128$	$[1230 + (\text{máx. pods en un nodo} - 64) \times 7.80] \text{ MB}$
	$\text{Máx. pods en un nodo} > 128$	$[1740 + (\text{máx. pods en un nodo} - 128) \times 11.20] \text{ MB}$

**AVISO**

Para un nodo de pequeña capacidad, ajuste el número máximo de instancias en función de los requisitos del sitio. Como alternativa, al crear un nodo en la consola de CCE, puede ajustar el número máximo de instancias para el nodo en función de las especificaciones del nodo.

## Reglas para reservar memoria de nodo (v2)

Para grupos de **v1.21.4-r0**, **v1.23.3-r0** o posteriores, el modelo de reserva de memoria de nodo se optimiza a V2 y se puede ajustar dinámicamente usando los parámetros de grupo de nodos **kube-reserved-mem** y **system-reserved-mem**. Para obtener más información, véase [Gestión de un grupo de nodos](#).

La memoria de nodo reservado total del modelo V2 es igual a la suma de la reservada para el SO y la reservada para que CCE gestione los pods.

La memoria reservada incluye las partes básicas y flotantes. Para el sistema operativo, la memoria flotante depende de las especificaciones del nodo. Para CCE, la memoria flotante depende del número de pods en un nodo.

**Tabla 3-14** Reglas para reservar memoria de nodo (v2)

Reservado para	Básico/flotante	Reserva	Utilizada por
SO	Básicos	400 MB (fijo)	Componentes de servicio del sistema operativo como sshd y systemd-journald.
	Flotante (dependiendo de la memoria del nodo)	25MB/GB	Kernel
CCE	Básicos	500 MB (fijo)	Componentes del motor de contenedores, como kubelet y kube-proxy, cuando el nodo está descargado
	Flotante (dependiendo del número de pods en el nodo)	Docker: 20 MB/pod containerd: 5 MB/pod	Componentes del motor de contenedores cuando aumenta el número de pods <b>NOTA</b> Cuando el modelo v2 reserva memoria para un nodo de forma predeterminada, el número máximo predeterminado de pods se estima en función de la memoria. Para obtener más información, véase <a href="#">Número máximo predeterminado de pods en un nodo.</a>

## Reglas para reservar CPU de nodo

**Tabla 3-15** Reglas de reserva de CPU de nodo

Núcleos totales de la CPU (Total)	Núcleos de CPU reservados
Total ≤ 1 núcleo	Total x 6%
1 núcleo < Total ≤ 2 núcleos	1 núcleo x 6% + (Total - 1 núcleo) x 1%
2 núcleos < Total ≤ 4 núcleos	1 núcleo x 6% + 1 núcleo x 1% + (Total - 2 núcleos) x 0.5%
Total > 4 núcleos	1 núcleo x 6% + 1 núcleo x 1% + 2 núcleos x 0.5% + (Total - 4 núcleos) x 0.25%

## Número máximo predeterminado de pods en un nodo

**Tabla 3-16** Número máximo predeterminado de pods en un nodo

Memoria	Número máximo predeterminado de pods
4 GB	20
8 GB	40
16 GB	60
32 GB	80
64 GB o más	110

### 3.1.6 Contenedores de Kata y contenedores comunes

La diferencia más significativa es que cada contenedor de Kata (pod) se ejecuta en una micro-VM independiente, tiene un núcleo de sistema operativo independiente y está aislado de forma segura en la capa de virtualización. CCE proporciona aislamiento contenedor más seguro que los clústeres privados independientes de Kubernetes. Con los núcleos del sistema operativo aislados, los recursos informáticos y las redes, los recursos y los datos de pods no serán evitados ni robados por otros pods.

Puede ejecutar un contenedor común o de Kata en un solo nodo en un clúster de CCE Turbo. La diferencia entre ellas es la siguiente:

Categoría	Contenedor de Kata	Contenedor común (Docker)	Contenedor común (containerd)
Tipo de nodo utilizado para ejecutar contenedores	Bare-metal server (BMS)	VM	VM
Motor de contenedores	containerd	Docker	containerd
Tiempo de ejecución de contenedores	Kata	runC	runC
Kernel de contenedores	Kernel exclusivo	Compartir el kernel con el host	Compartir el kernel con el host
Aislamiento de contenedores	VM livianas	cgroups y espacios de nombres	cgroups y espacios de nombres
Controlador de almacenamiento o del motor de contenedores	Device Mapper	OverlayFS2	OverlayFS

Categoría	Contenedor de Kata	Contenedor común (Docker)	Contenedor común (containerd)
<b>Sobrecarga del pod</b>	Memoria: 100 MiB CPU: 0.1 núcleos La sobrecarga de pod es una característica para contabilizar los recursos consumidos por la infraestructura de pod además de las solicitudes y límites de contenedor. Por ejemplo, si <b>limits.cpu</b> se ajusta a 0.5 núcleos y <b>limits.memory</b> a 256 MiB para un pod, el pod solicitará CPU de 0.6 núcleos y 356 MiB de memoria.	Ninguno	Ninguno
Especificaciones mínimas	Memoria: 256 MiB CPU: 0.25 núcleos Se recomienda que la relación de CPU (unidad: núcleo) a memoria (unidad: GiB) esté en el rango de 1:1 a 1:8. Por ejemplo, si la CPU tiene 0.5 núcleos, la memoria debería estar comprendida entre 512 MiB y 4 GiB.	Ninguno	Ninguno
CLI del motor de contenedores	crictl	Docker	crictl
Recursos informáticos de pod	Los valores de petición y límite deben ser los mismos tanto para la CPU como para la memoria.	Los valores de petición y límite pueden ser diferentes tanto para la CPU como para la memoria.	Los valores de petición y límite pueden ser diferentes tanto para la CPU como para la memoria.
Red de hosts	No se admite	Se admite	Se admite

Para obtener más información acerca de cómo utilizar los comandos de containerd y de Docker, consulte [Descripción del motor de contenedores](#).

### 3.1.7 Número máximo de pods que se pueden crear en un nodo

El número máximo de pods que se pueden crear en un nodo viene determinado por los siguientes parámetros:

- Número de direcciones IP de contenedor que se pueden asignar en un nodo (`alpha.cce/fixPoolMask`): Establezca este parámetro al crear un clúster de CCE. Este parámetro solo está disponible cuando **Network Model** tiene un valor de tipo **VPC network**.
- Número máximo de pods de un nodo (`maxPods`): Establezca este parámetro al crear un nodo. Es un concepto de configuración de kubelet.
- Número de ENI de un nodo de clúster de CCE Turbo: En un clúster de CCE Turbo, los nodos de ECS usan sub-ENI y los nodos de BMS usan ENI. El número máximo de pods que se pueden crear en un nodo depende del número de ENIs que puede utilizar el nodo.

El número máximo de pods que se pueden crear en un nodo depende del valor mínimo de estos parámetros.

- Para un clúster que utiliza el modelo de red de túnel de contenedor, el valor solo depende del **número máximo de pods de un nodo**.
- Para un clúster que utiliza el modelo de red VPC, el valor depende del **número máximo de pods en un nodo** y del **número de direcciones IP de contenedor que se pueden asignar a un nodo**. Se recomienda que el número máximo de pods en un nodo sea menor o igual al número de direcciones IP de contenedor que se pueden asignar al nodo. De lo contrario, es posible que los pods no se planifiquen cuando el número de direcciones IP de contenedor es insuficiente.
- Para un clúster (clúster de CCE Turbo) que utiliza el modelo Cloud Native Network 2.0, el valor depende del **número máximo de pods en un nodo** y del **número de NICs en un nodo de CCE Turbo**.

#### Red de contenedores vs. red de host

Al crear un pod, puede seleccionar la red de contenedor o la red host para el pod.

- Red de contenedores (predeterminada): **A cada pod se le asigna una dirección IP mediante los complementos de red del clúster, que ocupan las direcciones IP de la red de contenedor.**
- Red host: el pod utiliza la red host (**hostNetwork: true** necesita ser configurado para el pod) y ocupa el puerto host. La dirección IP del pod es la dirección IP del host. El pod no ocupa las direcciones IP de la red de contenedor. Para utilizar la red host, debe confirmar si los puertos de contenedor entran en conflicto con los puertos host. No utilice la red de host a menos que sepa exactamente qué puerto de host se utiliza por qué contenedor.

#### Número de direcciones IP de contenedor que se pueden asignar en un nodo

Si selecciona **VPC network** para **Network Model** al crear un clúster de CCE, también debe establecer el número de direcciones IP de contenedor que se pueden asignar a cada nodo.

Este parámetro afecta al número máximo de pods que se pueden crear en un nodo. Cada pod ocupa una dirección IP (cuando se utiliza la **red de contenedor**). Si el número de direcciones IP disponibles es insuficiente, no se pueden crear pods.

**Network Settings** Select the VPC and CIDR blocks for creating nodes and containers in the cluster.

Network Model VPC network Tunnel network [? Network Model Overview](#)

Model used for container networking in a cluster. Not editable after creation

Number of container IP addresses reserved for each node (cannot be changed after creation):  [Learn more](#)

De forma predeterminada, un nodo ocupa tres direcciones IP de contenedor (dirección de red, dirección de gateway y dirección de difusión). Por lo tanto, el número de direcciones IP de contenedor que se pueden asignar a un nodo es igual al número de direcciones IP de contenedor seleccionadas menos 3. Por ejemplo, en la figura anterior, **el número de direcciones IP de contenedor que se pueden asignar a un nodo es 125 (128 – 3)**.

## Número máximo de pods en un nodo

Al crear un nodo, puede configurar el número máximo de pods que se pueden crear en el nodo. This parameter is a configuration item of kubelet and determines the maximum number of pods that can be created by kubelet.

Max. Pods

Maximum number of pods (including default system pods) that can run properly on a node. This configuration prevents the node from

Each node in the current cluster can be allocated 125 container IP addresses for pods (specified when the cluster is created, e

## 3.2 Creación de un nodo

### Requisitos previos

- Se ha creado al menos un clúster.
- Se ha creado un par de claves para la autenticación de identidad al iniciar sesión en el nodo remoto.

Si utiliza una contraseña para iniciar sesión en un nodo, omita este paso. Para obtener más información, consulte [Creación de un par de claves](#).

### Restricciones

- El nodo tiene CPU de 2 núcleos o superior, 4 GB o más de memoria.
- Para garantizar la estabilidad del nodo, se reservará una cierta cantidad de recursos de nodo de CCE para los componentes de Kubernetes (como kubelet, kube-proxy y docker) en función de las especificaciones del nodo. Por lo tanto, el número total de recursos de nodo y recursos de nodo asignables en Kubernetes son diferentes. Cuanto más grandes sean las especificaciones del nodo, más se desplegará los contenedores en el nodo. Por lo tanto, es necesario reservar más recursos de nodo para ejecutar los componentes de Kubernetes. Para obtener más información, véase [Descripción de los recursos de nodos reservados](#).
- La red de nodos (como la red de VM y la red de contenedor) es tomada por CCE. No se le permite agregar y eliminar las NIC ni cambiar rutas. Si modifica la configuración de red, la disponibilidad de CCE puede verse afectada. Por ejemplo, la NIC llamada **gw\_11cbf51a@eth0** en el nodo es el gateway de red de contenedor y no se puede modificar.

- Si desea modificar las especificaciones de un nodo adquirido, detenga el nodo y realice las operaciones descritas en [Operaciones generales para modificar las especificaciones](#). También puede comprar un nuevo nodo y eliminar el antiguo.
- Durante la creación del nodo, los paquetes de software se descargan de OBS usando el nombre de dominio. Debe utilizar un servidor de DNS privado para resolver el nombre de dominio OBS y configurar la dirección del servidor de DNS de la subred donde reside el nodo con una [dirección del servidor DNS privado](#). Cuando se crea una subred, el servidor de DNS privado se utiliza de forma predeterminada. Si cambia el DNS de subred, asegúrese de que el servidor de DNS en uso puede resolver el nombre de dominio de OBS.
- Una vez que se crea un nodo, su AZ no se puede cambiar.
- Los nodos comprados en el modo de facturación de pago por uso se eliminarán después de eliminarlos en la página **Nodes** de la consola de CCE. **Yearly/monthly-billed nodes** en un clúster no se puede eliminar en la consola de CCE. Puede elegir **Billing Center > My Orders** en la esquina superior derecha de la página para cancelar la suscripción de los nodos.
- Services pueden verse comprometidos por los límites de ID de proceso de nodo. Es necesario evaluar si se debe ajustar el número máximo de PID. Para obtener más información, véase [Cambio de los límites de ID de proceso \(kernel.pid\\_max\)](#).
- Cuando la pila dual IPv4/IPv6 está habilitada, la concesión ilimitada DHCP no se puede habilitar para la subred de nodo seleccionada.

## Procedimiento

Después de crear un clúster, puede crear nodos para el clúster.

- Paso 1** Inicie sesión en la consola de CCE. En el panel de navegación, elija **Clusters**. Haga clic en el nombre del clúster de destino para acceder a su página de detalles.
- Paso 2** En el panel de navegación de la izquierda, elija **Nodes**. En la página que se muestra, haga clic en **Create Node**. En el paso **Node Settings**, establezca los parámetros de nodo haciendo referencia a la siguiente tabla.

### Ajustes de cómputo

Puede configurar las especificaciones y el sistema operativo de un servidor en la nube, en el que se ejecutan sus aplicaciones en contenedores.

**Tabla 3-17** Parámetros de configuración

Parámetro	Descripción
Modo de facturación	Se admiten los siguientes modos de facturación: <ul style="list-style-type: none"> <li>● Anual/Mensual                              Debe especificar la duración requerida si se selecciona <b>Yearly/Monthly</b>. Puede elegir si desea seleccionar <b>Auto-renew</b> según los requisitos del sitio. Su pedido se renovará automáticamente mensualmente o anualmente, dependiendo de si compró 1-9 meses, o 1-3 años.</li> <li>● Pago por uso                              Los recursos se facturarán en función de la duración del uso. Puede aprovisionar o eliminar recursos en cualquier momento.</li> </ul>

Parámetro	Descripción
AZ	<p>La zona de disponibilidad donde se encuentra el nodo. Los nodos de un clúster se pueden crear en las diferentes AZ para una mayor fiabilidad. El valor no se puede cambiar después de crear el nodo.</p> <p>Se recomienda seleccionar <b>Random</b> para desplegar su nodo en una AZ aleatoria basada en la variante de nodo seleccionado.</p> <p>Una AZ es una región física donde los recursos utilizan las fuentes de alimentación y las redes independientes. Las AZ están físicamente aisladas, pero se interconectan a través de una red interna. Para mejorar la disponibilidad de la carga de trabajo, cree nodos en las diferentes AZ.</p>
Node Type	<p>Clúster de CCE:</p> <ul style="list-style-type: none"> <li>● ECS (VM): Los contenedores se ejecutan en ECS.</li> <li>● ECS (físico): Los contenedores se ejecutan en servidores que utilizan la arquitectura QingTian.</li> <li>● BMS: Los contenedores se ejecutan en BMS. Es necesario adjuntar los discos locales o los discos de EVS.</li> </ul> <p>Clúster de CCE Turbo:</p> <ul style="list-style-type: none"> <li>● ECS (VM): Los contenedores se ejecutan en ECS. Solo ECS de Trunkport (modelos que se pueden unir con múltiples interfaces de red elástica (ENI)) son compatibles.</li> <li>● ECS (físico): Los contenedores se ejecutan en servidores que utilizan la arquitectura QingTian.</li> </ul>
Container Engine	<p>Los clústeres de CCE admiten Docker y containerd en algunos escenarios.</p> <ul style="list-style-type: none"> <li>● Los nodos que ejecutan CentOS, Ubuntu y EulerOS 2.9 soportan containerd. Los nodos de Arm que ejecutan EulerOS 2.5 y EulerOS 2.8 no admiten containerd.</li> <li>● Los clústeres de red de VPC de v1.23 y versiones posteriores admiten containerd. Los clústeres de red de túneles de contenedores de v1.23.2-r0 y versiones posteriores admiten containerd.</li> <li>● Para un clúster de CCE Turbo, <b>Docker</b> y <b>containerd</b> son compatibles. Para obtener más información, véase <a href="#">Asignación entre los sistemas operativos de nodos y los motores de contenedores</a>.</li> </ul>
Specifications	<p>Seleccione las especificaciones de nodo basadas en los requisitos de servicio. Las especificaciones de nodo disponibles varían en función de las AZ.</p> <p><b>NOTA</b></p> <ul style="list-style-type: none"> <li>● Los nodos de Kunpeng (con cómputo plus y con memoria optimizada) se pueden agregar a un clúster de CCE. Se utilizan las especificaciones reales que se muestran en la página de creación de nodos.</li> <li>● Actualmente, los clústeres de CCE Turbo no admiten los nodos de Kunpeng.</li> </ul>

Parámetro	Descripción
OS	<p>Select an OS type. Different types of nodes support different OSs.</p> <p><b>Public image:</b> Select an OS for the node.</p> <p><b>Private image:</b> You can use private images. Para obtener más información sobre cómo crear una imagen privada, consulte <a href="#">Creación de una imagen de nodo de CCE personalizada</a>.</p>
Node Name	<p>Nombre del nodo. Cuando los nodos (ECS) se crean por lotes, el valor de este parámetro se utiliza como prefijo de nombre para cada ECS.</p> <p>El sistema genera un nombre predeterminado para usted, que se puede modificar.</p> <p>Un nombre de nodo debe comenzar con una letra minúscula y no puede terminar con un guion (-). Solo se permiten dígitos, letras minúsculas y guiones (-).</p>
Login Mode	<ul style="list-style-type: none"> <li>● <b>Contraseña</b>                      El nombre de usuario predeterminado es <b>root</b>. Introduzca la contraseña para iniciar sesión en el nodo y confirme la contraseña.                      Asegúrese de recordar la contraseña, ya que la necesitará cuando inicie sesión en el nodo.</li> <li>● <b>Par de claves</b>                      Seleccione el par de claves utilizado para iniciar sesión en el nodo. Puede seleccionar una clave compartida.                      Se utiliza un par de claves para la autenticación de identidad cuando se inicia sesión de forma remota en un nodo. Si no hay ningún par de claves disponible, haga clic en <b>Create Key Pair</b>. Para obtener más información sobre cómo crear un par de claves, consulte <a href="#">Creación de un par de claves</a>.</li> </ul>

### Ajustes de almacenamiento

Configure los recursos de almacenamiento en un nodo para los contenedores que se ejecuta en él. Establezca el tamaño del disco según los requisitos del sitio.

**Tabla 3-18** Parámetros de configuración

Parámetro	Descripción
System Disk	<p>Disco del sistema utilizado por el sistema operativo del nodo. El valor oscila entre 40 GB y 1,024 GB. El valor predeterminado es 50 GB.</p> <p><b>Encryption:</b> La encriptación de disco de datos proporciona una protección potente para sus datos. Las instantáneas generadas a partir de discos cifrados y los discos creados con estas instantáneas heredan automáticamente la función de encriptación. <b>Esta función sólo está disponible en algunas regiones.</b></p> <ul style="list-style-type: none"><li>● <b>Encryption</b> no está seleccionado de forma predeterminada.</li><li>● Después de seleccionar <b>Encryption</b>, puede seleccionar una clave existente en el cuadro de diálogo que se muestra. Si no hay ninguna clave disponible, haga clic en <b>View Key List</b> para crear una clave. Una vez creada la clave, haga clic en el icono de actualización.</li></ul>

Parámetro	Descripción
Data Disk	<p><b>Se requiere al menos un disco de datos</b> para el tiempo de ejecución de contenedor y kubelet. <b>El disco de datos no se puede eliminar ni desinstalar. De lo contrario, el nodo no estará disponible.</b></p> <ul style="list-style-type: none"> <li>● Primer disco de datos: utilizado para el tiempo de ejecución de contenedor y kubelet. El valor oscila entre 20 GB y 32,768 GB. El valor predeterminado es 100 GB</li> <li>● Para otros discos de datos, el valor oscila entre 10 GB y 32,768 GB. El valor predeterminado es 100 GB.</li> </ul> <p><b>Configuración avanzada</b></p> <p>Haga clic en <b>Expand</b> para establecer los siguientes parámetros:</p> <ul style="list-style-type: none"> <li>● <b>Allocate Disk Space:</b> Seleccione esta opción para definir el espacio de disco ocupado por el tiempo de ejecución de contenedor para almacenar los directorios de trabajo, los datos de imagen de contenedor y los metadatos de imagen. Para obtener más información acerca de cómo asignar espacio en disco de datos, consulte <a href="#">Asignación de espacio en disco de datos</a>.</li> <li>● <b>Encryption:</b> La encriptación de disco de datos proporciona una protección potente para sus datos. Las instantáneas generadas a partir de discos cifrados y los discos creados con estas instantáneas heredan automáticamente la función de encriptación. <b>Esta función sólo está disponible en algunas regiones.</b> <ul style="list-style-type: none"> <li>– <b>Encryption</b> no está seleccionado de forma predeterminada.</li> <li>– Después de seleccionar <b>Encryption</b>, puede seleccionar una clave existente en el cuadro de diálogo que se muestra. Si no hay ninguna clave disponible, haga clic en <b>View Key List</b> para crear una clave. Una vez creada la clave, haga clic en el icono de actualización.</li> </ul> </li> </ul> <p><b>Adición de varios discos de datos</b></p> <p>Se puede agregar un máximo de cuatro discos de datos. De forma predeterminada, los discos sin procesar se crean sin ningún procesamiento. También puede hacer clic en <b>Expand</b> y seleccionar cualquiera de las siguientes opciones:</p> <ul style="list-style-type: none"> <li>● <b>Default:</b> De forma predeterminada, se crea un disco sin procesar sin ningún procesamiento.</li> <li>● <b>Mount Disk:</b> El disco de datos está conectado a un directorio especificado.</li> <li>● <b>Use as PV:</b> aplicable a escenarios en los que hay un requisito de alto rendimiento en PVs. La etiqueta <b>node.kubernetes.io/local-storage-persistent</b> se agrega al nodo con el PV configurado. El valor es <b>linear</b> o <b>striped</b>.</li> <li>● <b>Use as ephemeral volume:</b> aplicable a escenarios en los que EmptyDir exige un alto rendimiento.</li> </ul>

Parámetro	Descripción
	<p><b>NOTA</b></p> <ul style="list-style-type: none"> <li>● Los PV locales solo se admiten cuando la versión del clúster es v1.21.2-r0 o posterior y la versión adicional más reciente es 2.1.23 o posterior. Se recomienda 2.1.23 o posterior.</li> <li>● Los EV locales solo se admiten cuando la versión del clúster es v1.21.2-r0 o posterior y la versión adicional es 1.2.29 o posterior.</li> </ul> <p><b>PV local</b> y <b>EV local</b> soportan los siguientes modos de escritura:</p> <ul style="list-style-type: none"> <li>● <b>Linear</b>: Un volumen lógico lineal integra uno o más volúmenes físicos. Los datos se escriben en el siguiente volumen físico cuando se agota el anterior.</li> <li>● <b>Striped</b>: Un volumen lógico rayado separa los datos en bloques del mismo tamaño y los almacena en múltiples volúmenes físicos en secuencia, lo que permite que los datos se lean y escriban simultáneamente. No se puede ampliar un grupo de almacenamiento compuesto por volúmenes seccionados. Esta opción solo se puede seleccionar cuando existen varios volúmenes.</li> </ul> <p><b>Descripción de disco local</b></p> <p>Si la variante de nodo es con uso intensivo de disco o con capacidad ultraalta de E/S, un disco de datos puede ser un disco local.</p> <p>Los discos locales pueden descomponerse y no garantizar la fiabilidad de los datos. Se recomienda almacenar los datos de servicio en los discos de EVS, que son más fiables que los discos locales.</p>

### Ajustes de redes

Configure los recursos de red para permitir el acceso a nodos y aplicaciones en contenedores.

**Tabla 3-19** Parámetros de configuración

Parámetro	Descripción
Node Subnet	La subred de nodo seleccionada durante la creación del clúster se utiliza de forma predeterminada. Puede elegir otra subred en su lugar.
Node IP Address	Dirección IP del nodo especificado. De forma predeterminada, el valor se asigna aleatoriamente.
EIP	Un ECS sin la EIP vinculada no puede acceder a Internet ni ser accedido por redes públicas. El valor predeterminado es <b>Do not use</b> . <b>Use existing</b> y <b>Auto create</b> son compatibles.

### Configuración avanzada

Configure las capacidades avanzadas de nodo como etiquetas, manchas y comandos de inicio.

**Tabla 3-20** Parámetros de configuración avanzadas

Parámetro	Descripción
Kubernetes Label	<p>Haga clic en <b>Add Label</b> para establecer el par clave-valor asociado a los objetos de Kubernetes (como los pods). Se puede agregar un máximo de 20 etiquetas.</p> <p>Las etiquetas se pueden utilizar para distinguir nodos. Con la configuración de afinidad de carga de trabajo, los pods de contenedor se pueden programar en un nodo específico. Para obtener más información, consulte <a href="#">Etiquetas y selectores</a>.</p>
Resource Tag	<p>Puede agregar etiquetas de recursos para clasificar recursos.</p> <p>Puede crear <b>predefined tags</b> en Tag Management Service (TMS). Las etiquetas predefinidas son visibles para todos los recursos de servicio que admiten la función de etiquetado. Puede utilizar estas etiquetas para mejorar la eficiencia del etiquetado y la migración de recursos. Para obtener más información, consulte <a href="#">Creación de etiquetas predefinidas</a>.</p> <p>CCE creará automáticamente la etiqueta "CCE-Dynamic-Provisioning-Node=<i>node id</i>".</p>
Taint	<p>Este parámetro se deja en blanco por defecto. Puede agregar manchas para establecer antiafinidad para el nodo. Se permite un máximo de 10 manchas para cada nodo. Cada mancha contiene los siguientes parámetros:</p> <ul style="list-style-type: none"> <li>● <b>Key:</b> Una clave debe contener de 1 a 63 caracteres, comenzando por una letra o un dígito. Solo se permiten letras, dígitos, guiones (-), guiones bajos (_) y puntos (.). Un nombre de subdominio de DNS se puede utilizar como prefijo de una clave.</li> <li>● <b>Value:</b> Un valor debe comenzar con una letra o un dígito y puede contener un máximo de 63 caracteres, incluidos letras, dígitos, guiones (-), guiones bajos (_) y puntos (.).</li> <li>● <b>Effect:</b> Las opciones disponibles son <b>NoSchedule</b>, <b>PreferNoSchedule</b> y <b>NoExecute</b>.</li> </ul> <p>Para obtener más información, véase <a href="#">Gestión de manchas de nodos</a>.</p> <p><b>NOTA</b>                      Para un clúster de v1.19 o anterior, es posible que la carga de trabajo se haya programado en un nodo antes de agregar la mancha. Para evitar tal situación, seleccione un clúster de v1.19 o posterior.</p>
Max. Pods	<p>Número máximo de pods que se pueden ejecutar en el nodo, incluidos los pods del sistema predeterminados. Rango de valores: 16 a 256</p> <p>Este límite evita que el nodo se sobrecargue con pods.</p> <p>Este número también se decide por otros factores. Para obtener más información, véase <a href="#">Número máximo de pods que se pueden crear en un nodo</a>.</p>

Parámetro	Descripción
ECS Group	<p>Un grupo de ECS agrupa lógicamente ECS. Los ECS del mismo grupo de ECS cumplen con la misma política asociada con el grupo de ECS.</p> <p><b>Anti-affinity:</b> los ECS de un grupo de ECS se despliegan en diferentes hosts físicos para mejorar la confiabilidad del servicio.</p> <p>Seleccione un grupo de ECS existente o haga clic en <b>Add ECS Group</b> para crear uno. Una vez creado el grupo de ECS, haga clic en el botón de actualizar.</p>
Pre-installation Command	<p>Ingrese los comandos. Se permite un máximo de 1,000 caracteres.</p> <p>El script se ejecutará antes de que se instale el software de Kubernetes. Tenga en cuenta que si el script es incorrecto, es posible que el software de Kubernetes no se instale.</p>
Post-installation Command	<p>Ingrese los comandos. Se permite un máximo de 1,000 caracteres.</p> <p>El script se ejecutará después de instalar el software de Kubernetes y no afectará a la instalación.</p>
Agency	<p>El administrador de la cuenta crea una delegación en la consola de IAM. Al crear una delegación, puede compartir sus recursos de servidor en la nube con otra cuenta, o confiar a una persona o equipo más profesional para gestionar sus recursos.</p> <p>Si no hay ninguna delegación disponible, haga clic en <b>Create Agency</b> a la derecha para crear una.</p>

**Paso 3** Haga clic en **Next: Confirm**. Confirme los parámetros, especificaciones y tarifas configurados. Asegúrese de haber leído y comprendido la [Declaración del Image Management Service](#).

**Paso 4** Haga clic en **Submit**.

Si el nodo se facturará anualmente/mensualmente, haga clic en **Pay Now** y siga las instrucciones en pantalla para pagar el pedido.

Se muestra la página de lista de nodos. Si el estado del nodo es **Running**, el nodo se crea correctamente. Se tarda entre 6 y 10 minutos en crear un nodo.

**Paso 5** Haga clic en **Back to Node List**. El nodo se crea correctamente si cambia al estado **Running**.

----Fin

## Operaciones relacionadas

[Creación de un script de inyección de nodo.](#)

## 3.3 Adición de los nodos para gestión

### Escenario

En CCE, puede **Creación de un nodo** o agregar los nodos existentes (ECS/BMS) al clúster. Estos nodos se pueden **facturar en modo anual/mensual o de pago por uso**.

#### AVISO

- Mientras se acepta un ECS en un clúster, el sistema operativo del ECS se restablecerá a la imagen de sistema operativo estándar proporcionada por CCE para garantizar la estabilidad del nodo. La consola de CCE le pedirá que seleccione el sistema operativo y el modo de inicio de sesión durante el restablecimiento.
- El disco del sistema y el disco de datos de un ECS serán formateados mientras el ECS está siendo aceptado en un clúster. Asegúrese de que se ha realizado una copia de respaldo de la información de los discos.
- Mientras se acepta un ECS en un clúster, no realice ninguna operación en el ECS con la consola de ECS.

### Restricciones

- La versión del clúster debe ser 1.15 o posterior.
- Los nodos de Kunpeng solo pueden gestionar por los clústeres de v1.19 a v1.23.
- Puede gestionar los nodos de ECS, de BMS y de DeH, pero no los de HECS.
- Si **IPv6** está habilitado para un clúster, solo se pueden aceptar y gestionar los nodos de una subred con IPv6 habilitado. Si **IPv6** no está habilitado para el clúster, solo se pueden aceptar nodos de una subred sin IPv6 habilitado.
- Si se ha establecido la contraseña o la clave cuando se crea un nodo de máquina virtual, el nodo de máquina virtual se puede aceptar en un clúster 10 minutos después de que esté disponible. Durante la gestión, la contraseña o clave original no será válida. Necesita restablecer la contraseña o la clave.
- Los nodos de un clúster de CCE Turbo deben admitir sub-ENI o estar vinculados a al menos 16 ENI. Para obtener más información sobre las especificaciones de nodo, consulte los nodos que se pueden seleccionar en la consola al crear un nodo.
- El sistema operativo Ubuntu no es compatible cuando se gestionan los nodos de BMS.

### Requisitos previos

Se puede aceptar un servidor en la nube que cumpla las siguientes condiciones:

- El nodo que se aceptará debe estar en estado **Running** y no ser utilizado por otros clústeres. Además, el nodo que se va a aceptar no lleva la etiqueta de nodo de aprovisionamiento dinámico de CCE.
- El nodo que se aceptará y el clúster deben estar en la misma VPC. (Si la versión del clúster es anterior a v1.13.10, el nodo que se aceptará y el clúster de CCE deben estar en la misma subred.)

- Al menos un disco de datos está unido al nodo que se va a aceptar. La capacidad del disco de datos es mayor o igual a 100 GB. Para obtener más información sobre cómo conectar un disco de datos, consulte [Agregar un disco a un ECS](#).
- El nodo que se aceptará tiene una CPU de 2 núcleos o superior, 4 GB de memoria o más y solo una NIC.
- Si se utiliza un proyecto de empresa, el nodo que se aceptará y el clúster deben estar en el mismo proyecto de empresa. De lo contrario, los recursos no se pueden identificar durante la gestión. Como resultado, el nodo no se puede gestionar.
- Solo se pueden agregar por lotes los servidores en la nube con las mismas especificaciones, AZ y configuración de disco de datos.

## Procedimiento

- Paso 1** Inicie sesión en la consola de CCE y vaya al clúster donde reside el nodo que se va a gestionar.
- Paso 2** En el panel de navegación, elija **Nodes**. En la página mostrada, haga clic en **Accept Node** en la esquina superior derecha.
- Paso 3** Especifique los parámetros del nodo.

### Ajustes de cómputo

**Tabla 3-21** Parámetros de configuración

Parámetro	Descripción
Specifications	<p>Haga clic en <b>Select Cloud Server</b> y seleccione los servidores que desea aceptar.</p> <p>Puede seleccionar varios servidores en la nube para la gestión por lotes. Sin embargo, solo los servidores en la nube con las mismas especificaciones, AZ y configuración de disco de datos se pueden agregar por lotes.</p> <p>Si un servidor en la nube contiene varios discos de datos, seleccione uno de ellos para el tiempo de ejecución de contenedor y kubelet.</p>
Container Engine	<p>Los clústeres de CCE admiten Docker y containerd en algunos escenarios.</p> <ul style="list-style-type: none"> <li>● Los nodos que ejecutan CentOS, Ubuntu y EulerOS 2.9 soportan containerd. Los nodos de Arm que ejecutan EulerOS 2.5 y EulerOS 2.8 no admiten containerd.</li> <li>● Los clústeres de red de VPC de v1.23 y versiones posteriores admiten containerd. Los clústeres de red de túneles de contenedores de v1.23.2-r0 y versiones posteriores admiten containerd.</li> <li>● Para un clúster de CCE Turbo, <b>Docker</b> y <b>containerd</b> son compatibles. Para obtener más información, véase <a href="#">Asignación entre los sistemas operativos de nodos y los motores de contenedores</a>.</li> </ul>
OS	<p><b>Public image:</b> Seleccione un sistema operativo para el nodo.</p> <p><b>Private image:</b> Puede usar las imágenes privadas. Para obtener más información sobre cómo crear una imagen privada, consulte <a href="#">Creación de una imagen de nodo de CCE personalizada</a>.</p>

Parámetro	Descripción
Login Mode	<ul style="list-style-type: none"> <li>● <b>Contraseña</b>                      El nombre de usuario predeterminado es <b>root</b>. Introduzca la contraseña para iniciar sesión en el nodo y confirme la contraseña. Asegúrese de recordar la contraseña, ya que la necesitará cuando inicie sesión en el nodo.</li> <li>● <b>Par de claves</b>                      Seleccione el par de claves utilizado para iniciar sesión en el nodo. Puede seleccionar una clave compartida.                       Se utiliza un par de claves para la autenticación de identidad cuando se inicia sesión de forma remota en un nodo. Si no hay ningún par de claves disponible, haga clic en <b>Create Key Pair</b>. Para obtener más información sobre cómo crear un par de claves, consulte <a href="#">Creación de un par de claves</a>.</li> </ul>

### Ajustes de almacenamiento

Configure los recursos de almacenamiento en un nodo para los contenedores que se ejecuta en él.

**Tabla 3-22** Parámetros de configuración

Parámetro	Descripción
System Disk	Utilice directamente el disco del sistema del servidor en la nube.
Data Disk	<p><b>Se requiere al menos un disco de datos</b> para el tiempo de ejecución de contenedor y kubelet. <b>El disco de datos no se puede eliminar ni desinstalar. De lo contrario, el nodo no estará disponible.</b></p> <p>Haga clic en <b>Expand</b> y seleccione <b>Allocate Disk Space</b> para definir el espacio de disco ocupado por el tiempo de ejecución contenedor para almacenar los directorios de trabajo, los datos de imagen contenedor y los metadatos de imagen. Para obtener más información acerca de cómo asignar espacio en disco de datos, consulte <a href="#">Asignación de espacio en disco de datos</a>.</p> <p>Para otros discos de datos, se crea un disco sin procesar de forma predeterminada. También puede hacer clic en <b>Expand</b> y seleccionar <b>Mount Disk</b> para montar el disco de datos en un directorio especificado. Los discos de datos también se pueden usar para almacenamiento permanente o temporal. Para más detalles, véase <a href="#">PV local</a> y <a href="#">EV local</a>.</p>

### Configuración avanzada

**Tabla 3-23** Parámetros de configuración avanzadas

Parámetro	Descripción
Kubernetes Label	<p>Haga clic en <b>Add Label</b> para establecer el par clave-valor asociado a los objetos de Kubernetes (como los pods). Se puede agregar un máximo de 20 etiquetas.</p> <p>Las etiquetas se pueden utilizar para distinguir nodos. Con la configuración de afinidad de carga de trabajo, los pods de contenedor se pueden programar en un nodo específico. Para obtener más información, consulte <a href="#">Etiquetas y selectores</a>.</p>
Resource Tag	<p>Puede agregar etiquetas de recursos para clasificar recursos.</p> <p>Puede crear <b>etiquetas predefinidas</b> en Tag Management Service (TMS). Las etiquetas predefinidas son visibles para todos los recursos de servicio que admiten la función de etiquetado. Puede utilizar estas etiquetas para mejorar la eficiencia del etiquetado y la migración de recursos. Para obtener más información, consulte <a href="#">Creación de etiquetas predefinidas</a>.</p> <p>CCE creará automáticamente la etiqueta "CCE-Dynamic-Provisioning-Node=<i>node id</i>".</p>
Taint	<p>Este campo se deja en blanco por defecto. Puede agregar manchas para establecer antiafinidad para el nodo. Se permite un máximo de 10 manchas para cada nodo. Cada mancha contiene los siguientes parámetros:</p> <ul style="list-style-type: none"> <li>● <b>Key:</b> Una clave debe contener de 1 a 63 caracteres y comenzar por una letra o un dígito. Solo se permiten letras, dígitos, guiones (-), guiones bajos (_) y puntos (.). Un nombre de subdominio de DNS se puede utilizar como prefijo de una clave.</li> <li>● <b>Value:</b> Un valor debe comenzar con una letra o un dígito y puede contener un máximo de 63 caracteres, incluidos letras, dígitos, guiones (-), guiones bajos (_) y puntos (.).</li> <li>● <b>Effect:</b> Las opciones disponibles son <b>NoSchedule</b>, <b>PreferNoSchedule</b> y <b>NoExecute</b>.</li> </ul> <p><b>AVISO</b></p> <ul style="list-style-type: none"> <li>● Si se usan manchas, debe configurar tolerancias en los archivos YAML de pods. De lo contrario, la ampliación puede fallar o los pods no se pueden programar en los nodos agregados.</li> <li>● Después de crear un grupo de nodos, puede hacer clic en <b>Edit</b> para modificar su configuración. La modificación se sincronizará con todos los nodos del grupo de nodos.</li> </ul>
Max. Pods	<p>Número máximo de pods que se pueden ejecutar en el nodo, incluidos los pods del sistema predeterminados. Rango de valores: 16 a 256</p> <p>Este límite evita que el nodo se sobrecargue con pods.</p>
Pre-installation Command	<p>Ingrese los comandos. Se permite un máximo de 1,000 caracteres.</p> <p>El script se ejecutará antes de que se instale el software de Kubernetes. Tenga en cuenta que si el script es incorrecto, es posible que el software de Kubernetes no se instale.</p>

Parámetro	Descripción
Post-installation Command	Ingrese los comandos. Se permite un máximo de 1,000 caracteres. El script se ejecutará después de instalar el software de Kubernetes y no afectará a la instalación.

**Paso 4** Haga clic en **Next: Confirm**. Asegúrese de haber leído y entendido la [Declaración del Image Management Service](#). Haga clic en **Submit**.

----Fin

## 3.4 Inicio de sesión en un nodo

### Restricciones

- Si utiliza SSH para iniciar sesión en un nodo (un ECS), asegúrese de que el ECS ya tiene una EIP (una dirección IP pública).
- Solo se permite el inicio de sesión en un ECS en ejecución.
- Solo root del usuario puede iniciar sesión en un servidor Linux.

### Modos de inicio de sesión

Puede iniciar sesión en un ECS en cualquiera de los siguientes modos:

- Consola de gestión (VNC)  
 Si un ECS no tiene EIP, inicie sesión en la consola de ECS y haga clic en **Remote Login** en la misma fila que el ECS.  
 Para obtener más detalles, consulte [Inicio de sesión con VNC](#).
- SSH  
 Este modo solo se aplica a los ECS que ejecutan Linux. Por lo general, puede usar una herramienta de inicio de sesión remoto, como PuTTY, Xshell y SecureCRT, para iniciar sesión en su ECS. Si no se puede utilizar ninguna de las herramientas de inicio de sesión remoto, inicie sesión en la consola de ECS y haga clic en **Remote Login** en la misma fila que el ECS para ver el estado de conexión y el estado de ejecución del ECS.

#### NOTA

- Puede usar una clave de SSH o una contraseña de SSH para iniciar sesión. Para obtener más información, consulte [Inicio de sesión con una clave de SSH](#) e [Inicio de sesión con una contraseña de SSH](#).
- Cuando utilice el SO Windows para iniciar sesión en un nodo de Linux, establezca **Auto-login username** en root.
- La consola de CCE no admite la actualización del sistema operativo del nodo. No actualice el sistema operativo del nodo mediante el comando **yum update**. De lo contrario, los componentes de red contenedor no estarán disponibles. Para obtener más información sobre cómo restaurar manualmente la red contenedor consulte [¿Qué puedo hacer si la red de contenedores no está disponible después de la actualización yum se utiliza para actualizar el sistema operativo?](#)

**Tabla 3-24** Modos de inicio de sesión en un ECS de Linux

Vinculación de EIP	SO local	Método de conexión
Sí	Windows	Utilice una herramienta de inicio de sesión remoto, como PuTTY o XShell. <ul style="list-style-type: none"> <li>● Autenticación de contraseña SSH: <b>Iniciar sesión con una contraseña de SSH</b></li> <li>● Autenticación de clave de SSH: <b>Iniciar sesión con una clave de SSH</b></li> </ul>
Sí	Linux	Ejecute comandos. <ul style="list-style-type: none"> <li>● Autenticación de contraseña de SSH: <b>Iniciar sesión con una contraseña SSH</b></li> <li>● Autenticación de clave SSH: <b>Iniciar sesión con una clave de SSH</b></li> </ul>
Sí/No	Windows/Linux	Inicio de sesión remoto con la consola de gestión: <b>Iniciar sesión con VNC</b>

## 3.5 Nodos de gestión

### 3.5.1 Gestión de etiquetas de nodo

Puede agregar diferentes etiquetas a los nodos y definir diferentes atributos para las etiquetas. Al utilizar estas etiquetas de nodo, puede comprender rápidamente las características de cada nodo.

#### Escenario de uso de etiquetas de nodo.

Las etiquetas de nodo se utilizan principalmente en los siguientes escenarios:

- Gestión de nodos: Las etiquetas de nodo se utilizan para clasificar nodos.
- Afinidad y antiafinidad entre una carga de trabajo y un nodo:
  - Las diferentes cargas de trabajo tienen diferentes requisitos de recursos, como CPU, memoria y E/S. Si una carga de trabajo consume demasiados recursos en un clúster, es posible que otras cargas de trabajo del mismo clúster no se ejecuten correctamente. En este caso, se recomienda agregar diferentes etiquetas a los nodos. Al desplegar una carga de trabajo, puede seleccionar nodos con etiquetas específicas para el despliegue de afinidad para garantizar el funcionamiento normal del sistema. De lo contrario, se puede usar el despliegue antiafinidad de nodo.
  - Un sistema se puede dividir en múltiples módulos. Cada módulo consta de múltiples microservicios. Para garantizar una operación eficiente, puede agregar una etiqueta de módulo a cada nodo para que cada módulo pueda desplegarse en el nodo correspondiente. De esta manera, los módulos no interfieren entre sí y los microservicios pueden mantenerse fácilmente en sus nodos.

## Etiqueta inherente de un nodo

Después de crear un nodo, existen algunas etiquetas fijas y no se pueden eliminar. Para obtener más información sobre estas etiquetas, consulte [Tabla 3-25](#).

**Tabla 3-25** Etiqueta inherente de un nodo

Clave	Descripción
New: topology.kubernetes.io/ region Old: failure- domain.beta.kubernetes.io/ region	La región donde se encuentra el nodo
New: topology.kubernetes.io/ zone Old: failure- domain.beta.kubernetes.io/ zone	La zona de disponibilidad donde se encuentra el nodo
New: node.kubernetes.io/ baremetal Old: failure- domain.beta.kubernetes.io/is- baremetal	Si el nodo es un nodo de metal desnudo <b>false</b> indica que el nodo no es un nodo de metal desnudo.
node.kubernetes.io/instance- type	Especificaciones del nodo
kubernetes.io/arch	Arquitectura del procesador de nodos
kubernetes.io/hostname	Nombre del nodo
kubernetes.io/os	Tipo de SO
node.kubernetes.io/subnetid	ID de la subred donde se encuentra el nodo.
os.architecture	Arquitectura del procesador de nodos Por ejemplo, <b>amd64</b> indica un procesador de AMD64-bit.
os.name	Nombre del SO del nodo
os.version	Versión del kernel del SO de nodo
node.kubernetes.io/ container-engine	Motor de contenedores utilizado por el nodo.
accelerator/huawei-npu	Etiquetas de nodo de NPU.
accelerator	Etiquetas de nodo de GPU.
cce.cloud.com/cce-nodepool	Etiqueta dedicada de un nodo en un grupo de nodos.

## Adición o eliminación de una etiqueta de nodo

**Paso 1** Inicie sesión en la consola de CCE.

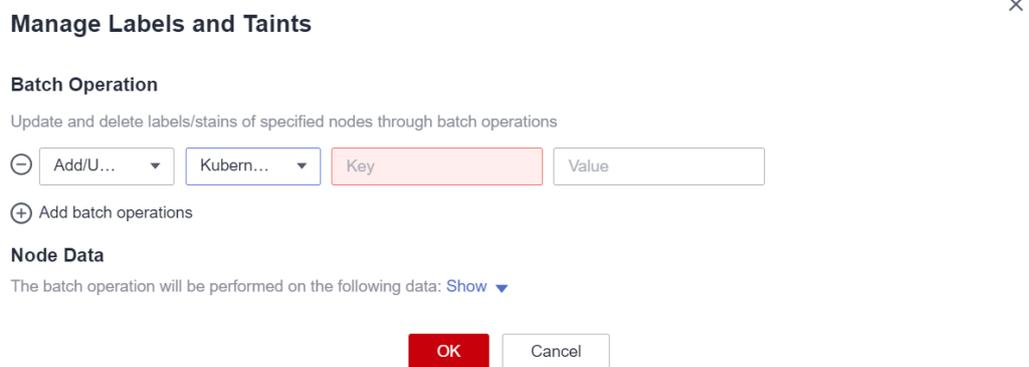
**Paso 2** Haga clic en el nombre del clúster, acceda a la página de detalles del clúster y elija **Nodes** en el panel de navegación. En la página mostrada, seleccione un nodo y haga clic en **Manage Labels and Taints**.

**Paso 3** En el cuadro de diálogo mostrado, haga clic en **Add batch operations** en **Batch Operation** y, a continuación, elija **Add/Update** o **Delete**.

Introduzca la clave y el valor de la etiqueta que desea agregar o eliminar y haga clic en **OK**.

Por ejemplo, la clave es **deploy\_qa** y el valor es **true** que indica que el nodo se usa para desplegar el entorno de QA (prueba).

**Figura 3-1** Adición de una etiqueta de nodo



**Paso 4** Después de agregar la etiqueta, compruebe la etiqueta agregada en los datos de nodo.

----Fin

## 3.5.2 Gestión de manchas de nodos

Las etiquetas permiten que un nodo repela los pods específicos para evitar que estos pods se programen en el nodo.

### Alteraciones

Una mancha es un par de clave y valor asociado con un efecto. Los siguientes efectos están disponibles:

- **NoSchedule**: Ningún pod podrá programar en el nodo a menos que tenga una tolerancia coincidente. Los pods existentes no serán desalojados del nodo.
- **PreferNoSchedule**: Kubernetes evita que los pods que no pueden tolerar esta contaminación se programen en el nodo.
- **NoExecute**: Si el pod se ha estado ejecutando en un nodo, el pod será desalojado del nodo. Si el pod no se ha ejecutado en un nodo, el pod no se programará en el nodo.

Para agregar una manch a un nodo, ejecute el comando **kubectl taint node *nodename*** de la siguiente manera:

```
$ kubectl get node
NAME          STATUS    ROLES    AGE    VERSION
```

```
192.168.10.170 Ready <none> 73d v1.19.8-r1-CCE21.4.1.B003
192.168.10.240 Ready <none> 4h8m v1.19.8-r1-CCE21.6.1.2.B001
$ kubectl taint node 192.168.10.240 key1=value1:NoSchedule
node/192.168.10.240 tainted
```

Para ver la configuración de mancha, ejecute los comandos **describe** y **get** de la siguiente manera:

```
$ kubectl describe node 192.168.10.240
Name: 192.168.10.240
...
Taints: key1=value1:NoSchedule
...
$ kubectl get node 192.168.10.240 -oyaml
apiVersion: v1
...
spec:
  providerID: 06a5ea3a-0482-11ec-8e1a-0255ac101dc2
  taints:
  - effect: NoSchedule
    key: key1
    value: value1
...
```

Para quitar una mancha, ejecute el siguiente comando con un guion (-) agregado después de **NoSchedule**:

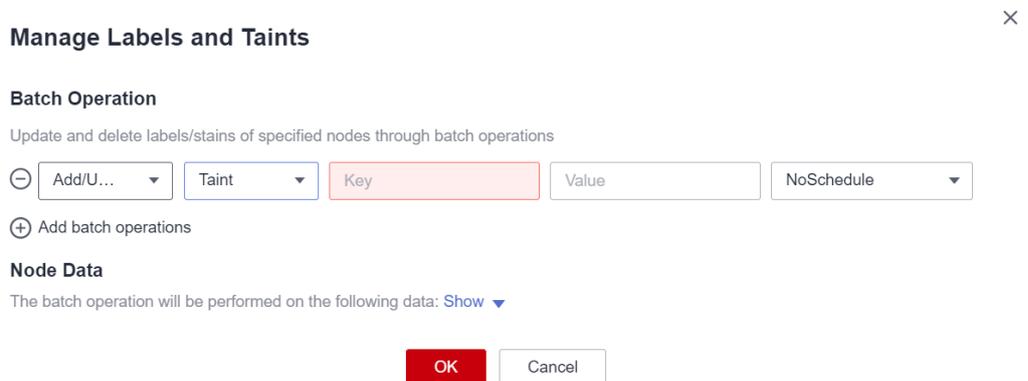
```
$ kubectl taint node 192.168.10.240 key1=value1:NoSchedule-
node/192.168.10.240 untainted
$ kubectl describe node 192.168.10.240
Name: 192.168.10.240
...
Taints: <none>
...
```

En la consola de CCE, también puede gestionar manchas de un nodo en lotes.

- Paso 1** Inicie sesión en la consola de CCE.
- Paso 2** Haga clic en el nombre del clúster, acceda a la página de detalles del clúster y elija **Nodes** en el panel de navegación. En la página mostrada, seleccione un nodo y haga clic en **Manage Labels and Taints**.
- Paso 3** En el cuadro de diálogo que se muestra, haga clic en **Add batch operations** en **Batch Operation**, elija **Add/Update** y seleccione **Taint**.

Ingrese la clave y el valor de la mancha que se agregará, seleccione el efecto de la mancha y haga clic en **OK**.

**Figura 3-2** Adición de una mancha



**Paso 4** Después de agregar la mancha, compruebe la mancha agregada en los datos del nodo.

----Fin

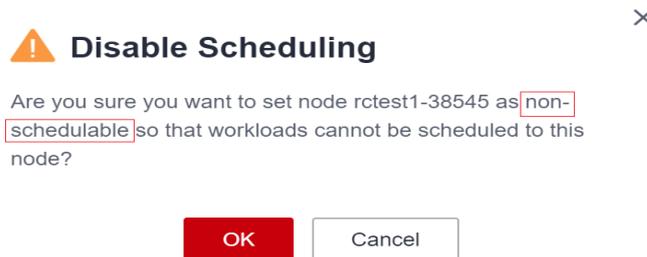
## Configuración de programación de nodos

Para configurar el ajuste de programación, inicie sesión en la consola de CCE, haga clic en el clúster, elija **Nodos** en el panel de navegación y haga clic en **More > Disable Scheduling** en la columna **Operation** de un nodo de la lista de nodos.

IP Address <sup>?</sup>	Pods (Allocated/To...)	CPU Request/Limit	Memory Request/Limit	Runtime Version & OS Version	Billing Mode	Operation
192.168.115.23...	5 / 110	30.61% ----- 30.61%	36.32% ----- 36.32%	docker://18.9.0 EulerOS 2.0 (SP5)	Pay-per-use Feb 09, 2022 14:24:23	Monitor   Sync ECS Data   More <sup>▲</sup>
192.168.115.20...	5 / 110	30.61% ----- 30.61%	36.32% ----- 36.32%	docker://18.9.0 EulerOS 2.0 (SP5)	Pay-per-use Feb 09, 2022 14:24:23	Monitor   S
192.168.113.72 ...	2 / 110	5.1% ----- 5.1%	9.88% ----- 9.88%	docker://18.9.0 EulerOS 2.0 (SP5)	Pay-per-use Feb 10, 2022 10:24:35	Monitor   S

- Event
- Pods
- View YAML
- Reset Node
- Disable Scheduling
- Change Billing Mode
- Remove
- Delete

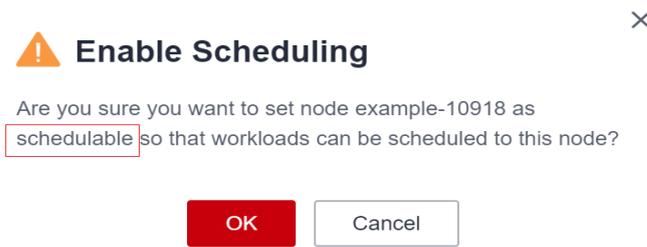
En el cuadro de diálogo que se muestra, haga clic en **OK** para establecer que el nodo no se puede programar.



Esta operación agregará una mancha al nodo. Puede usar kubectl para ver el contenido de la mancha.

```
$ kubectl describe node 192.168.10.240
...
Taints:          node.kubernetes.io/unschedulable:NoSchedule
...
```

En la consola de CCE, realice las mismas operaciones de nuevo para quitar la mancha y establecer el nodo para que sea programable.



## Tolerancias

Las tolerancias se aplican a los pods y permiten (pero no requieren) que los pods se planifiquen en nodos con las manchas coincidentes.

Las indicaciones y las tolerancias trabajan juntas para garantizar que los pods no estén programados en nodos inapropiados. Se aplican una o más manchas a un nodo. Esto marca que el nodo no debe aceptar ningún pod que no tolere las manchas.

Aquí hay un ejemplo de un pod que usa tolerancias:

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  labels:
    env: test
spec:
  containers:
  - name: nginx
    image: nginx
    imagePullPolicy: IfNotPresent
  tolerations:
  - key: "key1"
    operator: "Equal"
    value: "value1"
    effect: "NoSchedule"
```

En el ejemplo anterior, la etiqueta de tolerancia del pod es clave1=valor1 y el efecto de olor es de NoSchedule. Por lo tanto, el pod puede planificarse en el nodo correspondiente.

También puede configurar tolerancias similares a la siguiente información, que indica que el pod se puede programar en un nodo cuando el nodo tiene la mancha key1:

```
tolerations:
- key: "key1"
  operator: "Exists"
  effect: "NoSchedule"
```

### 3.5.3 Restablecimiento de un nodo

#### Escenario

Puede restablecer un nodo para modificar la configuración del nodo, como el sistema operativo del nodo y el modo de inicio de sesión.

Al restablecer un nodo se reinstala el sistema operativo del nodo y el software de Kubernetes en el nodo. Si un nodo no está disponible porque modifica la configuración del nodo, puede restablecer el nodo para que rectifique el error.

#### Restricciones

- Para los clústeres de CCE y de CCE Turbo, la versión debe ser v1.13 o posterior para admitir el restablecimiento de nodos.
- Para los clústeres de Kunpeng, la versión debe ser v1.15 o posterior para admitir el restablecimiento de nodos.

#### Notas

- Solo se pueden restablecer los nodos de trabajo. Si el nodo aún no está disponible después del restablecimiento, elimine el nodo y cree uno nuevo.

- **Al restablecer un nodo se reinstala el SO del nodo e interrumpe los servicios de carga de trabajo que se ejecutan en el nodo. Por lo tanto, realice esta operación durante las horas fuera de pico.**
- **Los datos en el disco del sistema y los discos de datos de Docker se borrarán. Haga una copia de seguridad de los datos importantes antes de restablecer el nodo.**
- **Cuando se monta un disco de datos adicional en un nodo, los datos de este disco se borrarán si el disco no se ha desmontado antes de que se restablezca el nodo. Para evitar la pérdida de datos, realice una copia de seguridad de los datos por adelantado y vuelva a montar el disco de datos después de que se complete el restablecimiento del nodo.**
- Las direcciones IP de los pods de carga de trabajo en el nodo cambiarán, pero el acceso a la red del contenedor no se ve afectado.
- Hay una cuota de disco de EVS restante.
- Mientras se elimina el nodo, el backend establecerá el nodo en el estado no programado.
- El restablecimiento de un nodo provocará la pérdida de datos de PVC/PV para los **PV locales** asociados con el nodo. Estos PVC y PV no se pueden restaurar o utilizar de nuevo. En este escenario, el pod que utiliza el PV local se desaloja del nodo de reinicio. Se crea un nuevo pod y permanece en el estado pendiente. Esto se debe a que el PVC utilizado por el pod tiene una etiqueta de nodo, debido a lo cual el pod no se puede programar. Después de que se restablezca el nodo, el pod puede planificarse para el nodo de reinicio. En este caso, la cápsula está siempre en el estado de creación porque el volumen lógico subyacente correspondiente al PVC no existe.

## Procedimiento

La nueva consola le permite restablecer nodos por lotes. También puede usar imágenes privadas para restablecer nodos por lotes.

**Paso 1** Inicie sesión en la consola de CCE.

**Paso 2** Haga clic en el nombre del clúster y acceda a la página de detalles del clúster, elija **Nodos** en el panel de navegación y seleccione uno o varios nodos para restablecer en la lista de la derecha. Elija **More > Reset**.

**Paso 3** En el cuadro de diálogo que se muestra, haga clic en **Yes**.

- Para los nodos del grupo de nodos DefaultPool, se muestra la página de configuración de parámetros. Establezca los parámetros haciendo referencia a **Paso 4**.
- Para un nodo que cree en un grupo de nodos, el restablecimiento del nodo no admite la configuración de parámetros. Puede utilizar directamente la imagen de configuración del grupo de nodos para restablecer el nodo.

**Paso 4** Especifique los parámetros del nodo.

### Ajustes de cómputo

**Tabla 3-26** Parámetros de configuración

Parámetro	Descripción
Specifications	Las especificaciones no se pueden modificar al restablecer un nodo.

Parámetro	Descripción
Container Engine	<p>Los clústeres de CCE admiten Docker y containerd en algunos escenarios.</p> <ul style="list-style-type: none"> <li>● Los nodos que ejecutan CentOS, Ubuntu y EulerOS 2.9 soportan containerd. Los nodos de Arm que ejecutan EulerOS 2.5 y EulerOS 2.8 no admiten containerd.</li> <li>● Los clústeres de red de VPC de v1.23 y versiones posteriores admiten containerd. Los clústeres de red de túneles de contenedores de v1.23.2-r0 y versiones posteriores admiten containerd.</li> <li>● Para un clúster de CCE Turbo, <b>Docker</b> y <b>containerd</b> son compatibles. Para obtener más información, véase <a href="#">Asignación entre los sistemas operativos de nodos y los motores de contenedores</a>.</li> </ul>
OS	<p><b>Public image:</b> Seleccione un sistema operativo para el nodo.</p> <p><b>Private image:</b> Puede usar las imágenes privadas. Para obtener más información sobre cómo crear una imagen privada, consulte <a href="#">Creación de una imagen de nodo de CCE personalizada</a>.</p>
Login Mode	<ul style="list-style-type: none"> <li>● <b>Contraseña</b>                      El nombre de usuario predeterminado es <b>root</b>. Introduzca la contraseña para iniciar sesión en el nodo y confirme la contraseña. Asegúrese de recordar la contraseña, ya que la necesitará cuando inicie sesión en el nodo.</li> <li>● <b>Par de claves</b>                      Seleccione el par de claves utilizado para iniciar sesión en el nodo. Puede seleccionar una clave compartida.                       Se utiliza un par de claves para la autenticación de identidad cuando se inicia sesión de forma remota en un nodo. Si no hay ningún par de claves disponible, haga clic en <b>Create Key Pair</b>. Para obtener más información sobre cómo crear un par de claves, consulte <a href="#">Creación de un par de claves</a>.</li> </ul>

### Ajustes de almacenamiento

Configure los recursos de almacenamiento en un nodo para los contenedores que se ejecuta en él.

**Tabla 3-27** Parámetros de configuración

Parámetro	Descripción
System Disk	Utilice directamente el disco del sistema del servidor en la nube.

Parámetro	Descripción
Data Disk	<p><b>Se requiere al menos un disco de datos</b> para el tiempo de ejecución de contenedor y kubelet. <b>El disco de datos no se puede eliminar ni desinstalar. De lo contrario, el nodo no estará disponible.</b></p> <p>Haga clic en <b>Expand</b> y seleccione <b>Allocate Disk Space</b> para definir el espacio de disco ocupado por el tiempo de ejecución contenedor para almacenar los directorios de trabajo, los datos de imagen contenedor y los metadatos de imagen. Para obtener más información acerca de cómo asignar espacio en disco de datos, consulte <a href="#">Asignación de espacio en disco de datos</a>.</p> <p>Para otros discos de datos, se crea un disco sin procesar de forma predeterminada. También puede hacer clic en <b>Expand</b> y seleccionar <b>Mount Disk</b> para montar el disco de datos en un directorio especificado. Los discos de datos también se pueden usar para almacenamiento permanente o temporal. Para más detalles, véase <a href="#">PV local</a> y <a href="#">EV local</a>.</p>

### Configuración avanzada

**Tabla 3-28** Parámetros de configuración avanzadas

Parámetro	Descripción
Kubernetes Label	<p>Haga clic en <b>Add Label</b> para establecer el par clave-valor asociado a los objetos de Kubernetes (como los pods). Se puede agregar un máximo de 20 etiquetas.</p> <p>Las etiquetas se pueden utilizar para distinguir nodos. Con la configuración de afinidad de carga de trabajo, los pods de contenedor se pueden programar en un nodo específico. Para obtener más información, consulte <a href="#">Etiquetas y selectores</a>.</p>
Resource Tag	<p>Puede agregar etiquetas de recursos para clasificar recursos.</p> <p>Puede crear <b>etiquetas predefinidas</b> en Tag Management Service (TMS). Las etiquetas predefinidas son visibles para todos los recursos de servicio que admiten la función de etiquetado. Puede utilizar estas etiquetas para mejorar la eficiencia del etiquetado y la migración de recursos. Para obtener más información, consulte <a href="#">Creación de etiquetas predefinidas</a>.</p> <p>CCE creará automáticamente la etiqueta "CCE-Dynamic-Provisioning-Node=<i>node id</i>".</p>

Parámetro	Descripción
Taint	<p>Este campo se deja en blanco por defecto. Puede agregar manchas para establecer antiafinidad para el nodo. Se permite un máximo de 10 manchas para cada nodo. Cada mancha contiene los siguientes parámetros:</p> <ul style="list-style-type: none"> <li>● <b>Key:</b> Una clave debe contener de 1 a 63 caracteres y comenzar por una letra o un dígito. Solo se permiten letras, dígitos, guiones (-), guiones bajos (_) y puntos (.). Un nombre de subdominio de DNS se puede utilizar como prefijo de una clave.</li> <li>● <b>Value:</b> Un valor debe comenzar con una letra o un dígito y puede contener un máximo de 63 caracteres, incluidos letras, dígitos, guiones (-), guiones bajos (_) y puntos (.).</li> <li>● <b>Effect:</b> Las opciones disponibles son <b>NoSchedule</b>, <b>PreferNoSchedule</b> y <b>NoExecute</b>.</li> </ul> <p><b>AVISO</b></p> <ul style="list-style-type: none"> <li>● Si se usan manchas, debe configurar tolerancias en los archivos YAML de pods. De lo contrario, la ampliación puede fallar o los pods no se pueden programar en los nodos agregados.</li> <li>● Después de crear un grupo de nodos, puede hacer clic en <b>Edit</b> para modificar su configuración. La modificación se sincronizará con todos los nodos del grupo de nodos.</li> </ul>
Max. Pods	<p>Número máximo de pods que se pueden ejecutar en el nodo, incluidos los pods del sistema predeterminados. Rango de valores: 16 a 256</p> <p>Este límite evita que el nodo se sobrecargue con pods.</p>
Pre-installation Command	<p>Ingrese los comandos. Se permite un máximo de 1,000 caracteres. El script se ejecutará antes de que se instale el software de Kubernetes. Tenga en cuenta que si el script es incorrecto, es posible que el software de Kubernetes no se instale.</p>
Post-installation Command	<p>Ingrese los comandos. Se permite un máximo de 1,000 caracteres. El script se ejecutará después de instalar el software de Kubernetes y no afectará a la instalación.</p>

**Paso 5** Haga clic en **Next: Confirm**. Asegúrese de haber leído y entendido la [Declaración del Image Management Service](#).

**Paso 6** Haga clic en **Submit**.

---Fin

## 3.5.4 Extracción de un nodo

### Escenario

Al quitar un nodo de un clúster se reinstala el sistema operativo del nodo y se borran los componentes de CCE en el nodo.

Quitar un nodo no eliminará el servidor correspondiente al nodo. Se recomienda quitar los nodos en horas fuera de pico para evitar impactos en sus servicios.

Después de eliminar un nodo del clúster, el nodo sigue ejecutando e incurre en las tarifas.

## Notas y restricciones

- Los nodos solo se pueden quitar cuando el clúster se encuentra en estado **Available** o **Unavailable**.
- Un nodo de CCE se puede quitar solo cuando está en el estado **Active**, **Abnormal** o **Error**.
- Un nodo de CCE en el estado **Active** puede volver a instalar su sistema operativo y borrar los componentes de CCE después de que se quite.
- Si el sistema operativo no se reinstala después de quitar el nodo, vuelva a instalarlo manualmente. Después de la reinstalación, inicie sesión en el nodo y ejecute el script de liquidación para borrar los componentes de CCE. Para obtener más información, véase [Manejo de la reinstalación fallida del sistema operativo](#).
- Quitar un nodo provocará la pérdida de datos de PVC/PV para el **PV local** asociado con el nodo. Estos PVC y PV no se pueden restaurar o utilizar de nuevo. En este escenario, el pod que utiliza el PV local se desaloja del nodo. Se crea un nuevo pod y permanece en el estado pendiente. Esto se debe a que el PVC utilizado por el pod tiene una etiqueta de nodo, debido a lo cual el pod no se puede programar.

## Precauciones

- La eliminación de un nodo conducirá a la migración de pods, lo que puede afectar a los servicios. Realice esta operación durante las horas de menor actividad.
- Pueden producirse riesgos inesperados durante la operación. Realice una copia de respaldo de los datos por adelantado.
- Mientras se elimina el nodo, el backend establecerá el nodo en el estado no programado.
- Después de quitar el nodo y volver a instalar el sistema operativo, las particiones de LVM originales se borrarán y los datos administrados por LVM se borrarán. Por lo tanto, realice una copia de respaldo de los datos por adelantado.

## Procedimiento

**Paso 1** Inicie sesión en la consola de CCE y haga clic en el nombre del clúster para acceder al clúster.

**Paso 2** Elija **Nodes** en el panel de navegación y elija **More > Remove** en la columna **Operation** del nodo de destino.

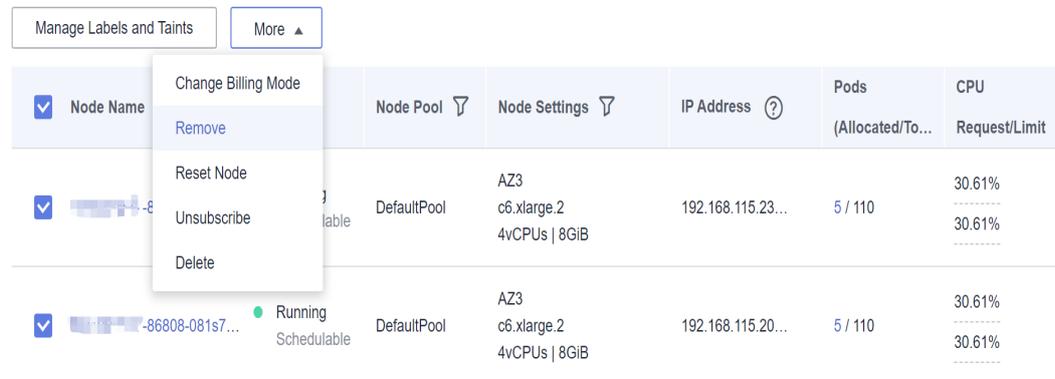
**Figura 3-3** Quitar un nodo

Node Settings	IP Address	Pods (Allocated/To...)	CPU Request/Limit	Memory Request/Limit	Runtime Version & OS Version	Billing Mode	Operation
AZ3 c6.xlarge.2 4vCPUs   8GiB	192.168.115.23...	5 / 110	30.61% -----	36.32% -----	docker://18.9.0 EulerOS 2.0 (SP5)	Pay-per-use Feb 09, 2022 14:24:23	Monitor   Sync ECS Data   More
AZ3 c6.xlarge.2 4vCPUs   8GiB	192.168.115.20...	5 / 110	30.61% -----	36.32% -----	docker://18.9.0 EulerOS 2.0 (SP5)	Pay-per-use Feb 09, 2022 14:24:23	Monitor   S
AZ3 c6.xlarge.2 4vCPUs   8GiB	192.168.113.72 ...	2 / 110	5.1% -----	9.88% -----	docker://18.9.0 EulerOS 2.0 (SP5)	Pay-per-use Feb 10, 2022 10:24:35	Monitor   S

- Event
- Pods
- View YAML
- Reset Node
- Disable Scheduling
- Change Billing Mode
- Remove
- Delete

También puede seleccionar varios nodos y quitarlos a la vez.

**Figura 3-4** Quitar varios nodos a la vez



**Paso 3** En el cuadro de diálogo que se muestra, configure la información de inicio de sesión necesaria para volver a instalar el sistema operativo y haga clic en **Yes**. Espere hasta que se quite el nodo.

Después de quitar el nodo, los pods de carga de trabajo del nodo se migran automáticamente a otros nodos disponibles.

----Fin

## Manejo de la reinstalación fallida del sistema operativo

Puede realizar los siguientes pasos para volver a instalar el sistema operativo y borrar los componentes de CCE en el nodo si fallan los intentos anteriores:

**Paso 1** Inicie sesión en la consola de gestión del servidor y vuelva a instalar el sistema operativo. Para obtener más información, consulte [Cambiar el sistema operativo](#).

**Paso 2** Inicie sesión en el servidor y ejecute los siguientes comandos para borrar los componentes de CCE y los datos de LVM:

Escriba las siguientes secuencias de comandos en el archivo **clean.sh**:

```
lsblk
vgs --noheadings | awk '{print $1}' | xargs vgremove -f
pvs --noheadings | awk '{print $1}' | xargs pvremove -f
lvs --noheadings | awk '{print $1}' | xargs -i lvremove -f --select {}
function init_data_disk() {
    all_devices=$(lsblk -o KNAME,TYPE | grep disk | grep -v nvme | awk '{print $1}' | awk '{ print "/dev/"$1}')
    for device in ${all_devices[@]}; do
        isRootDisk=$(lsblk -o KNAME,MOUNTPOINT $device 2>/dev/null | grep -E '[:space:]]/$' | wc -l)
        if [[ ${isRootDisk} != 0 ]]; then
            continue
        fi
        dd if=/dev/urandom of=${device} bs=512 count=64
        return
    done
    exit 1
}
init_data_disk
lsblk
```

Ejecute el siguiente comando:

```
bash clean.sh
```

```
----Fin
```

### 3.5.5 Sincronización de datos con servidores en la nube

#### Escenario

Cada nodo de un clúster es un servidor en la nube o una máquina física. Después de crear un nodo de clúster, puede cambiar el nombre o las especificaciones del servidor en la nube según sea necesario.

Alguna información sobre los nodos de CCE se mantiene independientemente de la consola de ECS. Después de cambiar el nombre, EIP, modo de facturación o especificaciones de un ECS en la consola de ECS, debe **sincronizar la información de ECS** con el nodo correspondiente en la consola de CCE. Después de la sincronización, la información en ambas consolas es consistente.

Información común de ECS que se modificará:

- Nombre de ECS (nodo): [Cambiar un nombre de ECS](#)
- Especificación de nodo: [Operaciones generales para la modificación de especificaciones](#)

#### Restricciones

- Los datos, incluidos el estado de la máquina virtual, los nombres de ECS, el número de CPU, el tamaño de la memoria, las especificaciones de ECS y las direcciones IP públicas, se pueden sincronizar.  
 Si se especifica un nombre de ECS como nombre de nodo de Kubernetes, el cambio del nombre de ECS no se puede sincronizar con la consola de CCE.
- Los datos, como el sistema operativo y el ID de imagen, no se pueden sincronizar. (Estos parámetros no se pueden modificar en la consola de ECS.)

#### Procedimiento

- Paso 1** Inicie sesión en la consola de CCE.
- Paso 2** Haga clic en el nombre del clúster para acceder a la consola del clúster. Elija **Nodes** en el panel de navegación.
- Paso 3** Elija **More > Sync Server Data** junto al nodo.

**Figura 3-5** Sincronización de datos del servidor

IP Address <sup>?</sup>	Pods (Allocated/To...)	CPU Request/Limit	Memory Request/Limit	Runtime Version & OS Version	Billing Mode	Operation
192.168.115.23...	5 / 110	30.61% -----	36.32% -----	docker://18.9.0 EulerOS 2.0 (SP5)	Pay-per-use Feb 09, 2022 14:24:23	Monitor   Sync ECS Data   More ▾
192.168.115.20...	5 / 110	30.61% -----	36.32% -----	docker://18.9.0 EulerOS 2.0 (SP5)	Pay-per-use Feb 09, 2022 14:...	

After modifying the server name, IP address, and specifications on the network console, you can click this button to synchronize the server data to the node.

Una vez completada la sincronización, aparece el mensaje **ECS data synchronization requested** en la esquina superior derecha.

----Fin

## 3.5.6 Eliminación de un nodo

### Escenario

Cuando se elimina un nodo de un clúster CCE, también se eliminan los servicios que se ejecutan en el nodo. Tenga cuidado al realizar esta operación.

### Notas y restricciones

- Si los nodos se facturan anualmente/mensualmente, no se pueden eliminar directamente. Puede elegir **Billing Center > My Orders** para darse de baja de los nodos.
- Los nodos de máquinas virtuales que utiliza CCE no admiten la cancelación de la suscripción ni la eliminación en la página de ECS.
- Para un clúster de v1.17.11 o posterior, después de que un servidor de metal puro se cancela o se elimina en la consola de BMS, el nodo correspondiente del clúster se elimina automáticamente.
- La eliminación de un nodo causará la pérdida de datos de PVC/PV para los **PV locales** asociados con el nodo. Estos PVC y PV no se pueden restaurar o utilizar de nuevo. En este escenario, el pod que utiliza el PV local se desaloja del nodo. Se crea un nuevo pod y permanece en el estado pendiente. Esto se debe a que el PVC utilizado por el pod tiene una etiqueta de nodo, debido a lo cual el pod no se puede programar.

### Precauciones

- La eliminación de un nodo conducirá a la migración de pods, que puede afectar a los servicios. Realice esta operación durante las horas de menor actividad.
- Pueden producirse riesgos inesperados durante la operación. Haga una copia de respaldo de los datos relacionados con anticipación.
- Durante la operación, el backend establecerá el nodo en el estado no programado.
- Solo se pueden eliminar los nodos de trabajo.

### Procedimiento

**Paso 1** Inicie sesión en la consola de CCE y haga clic en el nombre del clúster para acceder al clúster.

**Paso 2** En el panel de navegación, elija **Nodes**. En la misma fila que el nodo que eliminará, elija **More > Delete**.

**Paso 3** En el cuadro de diálogo **Delete Node**, haga clic en **Yes**.

#### NOTA

- Después de eliminar el nodo, los pods de él se migran automáticamente a otros nodos disponibles.
- Si los discos y las EIP unidas al nodo son recursos importantes, desvínculelos primero. De lo contrario, se eliminarán con el nodo.

----Fin

## 3.5.7 Cambio de pago por uso a anual/mensual

CCE admite la facturación de **pay-per-use** y **yearly/monthly**. Un nodo de pago por uso se puede cambiar a facturado anual/mensual.

### Restricciones

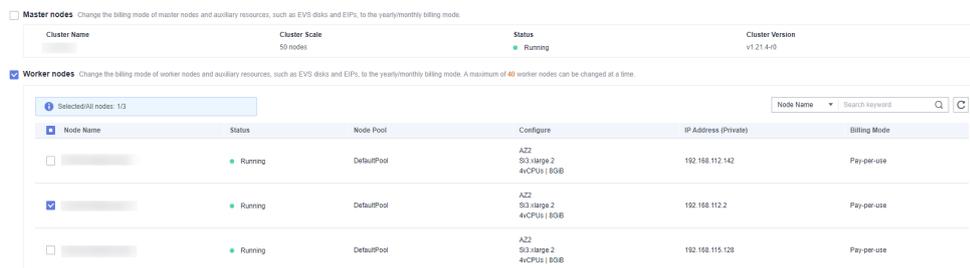
- No puede cambiar los nodos de pago por uso a anual/mensual en la consola de ECS.
- Solo los nodos del grupo de nodos predeterminado **DefaultPool** se pueden cambiar al modo de facturación anual/mensual.
- Los nodos cuyo modo de facturación se cambia a anual/mensual no admiten el ajuste automático.

### Cambio a la facturación anual/mensual

Realice los siguientes pasos:

- Paso 1** Inicie sesión en la consola de CCE y haga clic en el nombre del clúster para acceder al clúster.
- Paso 2** En el panel de navegación de la izquierda, elija **Nodos**. En la fila del nodo de destino, elija **More > Change Billing Mode**.
- Paso 3** En la página **Change Billing Mode**, elija los nodos que se cambiarán a anual/mensual.

**Figura 3-6** Cambio de los modos de facturación para los nodos principal y trabajador



- Paso 4** Haga clic en **OK**. Espere hasta que se procese el pedido y se complete el pago.

Durante el pago, si aparece un mensaje indicando que **you do not have the permission to access the resource API** (Usted no tiene el permiso para acceder al recurso API), vuelva a la página anterior y vuelva a realizar la operación.

----Fin

## 3.5.8 Detención de un nodo

### Escenario

Después de detener un nodo en el clúster, los servicios en el nodo también se detienen. Antes de detener un nodo, asegúrese de que la discontinuidad de los servicios en el nodo no dará lugar a impactos adversos.

La mayoría de los nodos ya no se facturan después de detenerlos, excluidos ciertos tipos de ECS (unos con discos locales conectados, como ECS con uso intensivo de disco y con capacidad ultraalta de E/S). Para obtener más información, consulte [Facturación de ECS](#).

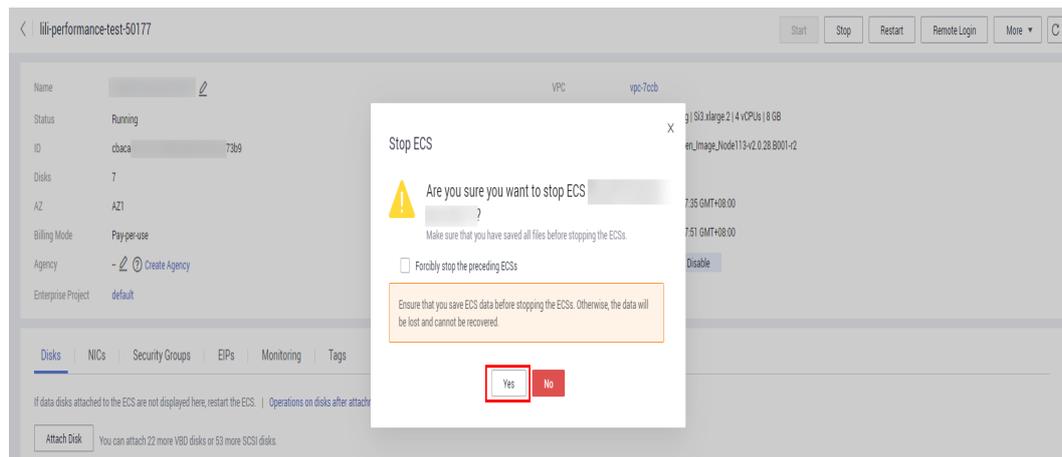
## Restricciones

- La eliminación de un nodo conducirá a la migración de pods, que puede afectar a los servicios. Por lo tanto, elimine los nodos durante las horas no pico.
- Pueden producirse riesgos inesperados durante la eliminación de nodos. Haga una copia de respaldo de los datos relacionados con anticipación.
- Mientras se elimina el nodo, el backend establecerá el nodo en el estado no programado.
- Solo se pueden detener los nodos de trabajo.

## Procedimiento

- Paso 1** Inicie sesión en la consola de CCE y haga clic en el nombre del clúster para acceder al clúster.
- Paso 2** En el panel de navegación, elija **Nodos**. En el panel derecho, haga clic en el nombre del nodo que desea detener.
- Paso 3** En la esquina superior derecha de la página de detalles de ECS, haga clic en **Stop** en el área de estado de la instancia. En el cuadro de diálogo que se muestra, haga clic en **Yes**.

**Figura 3-7** Página de detalles de ECS



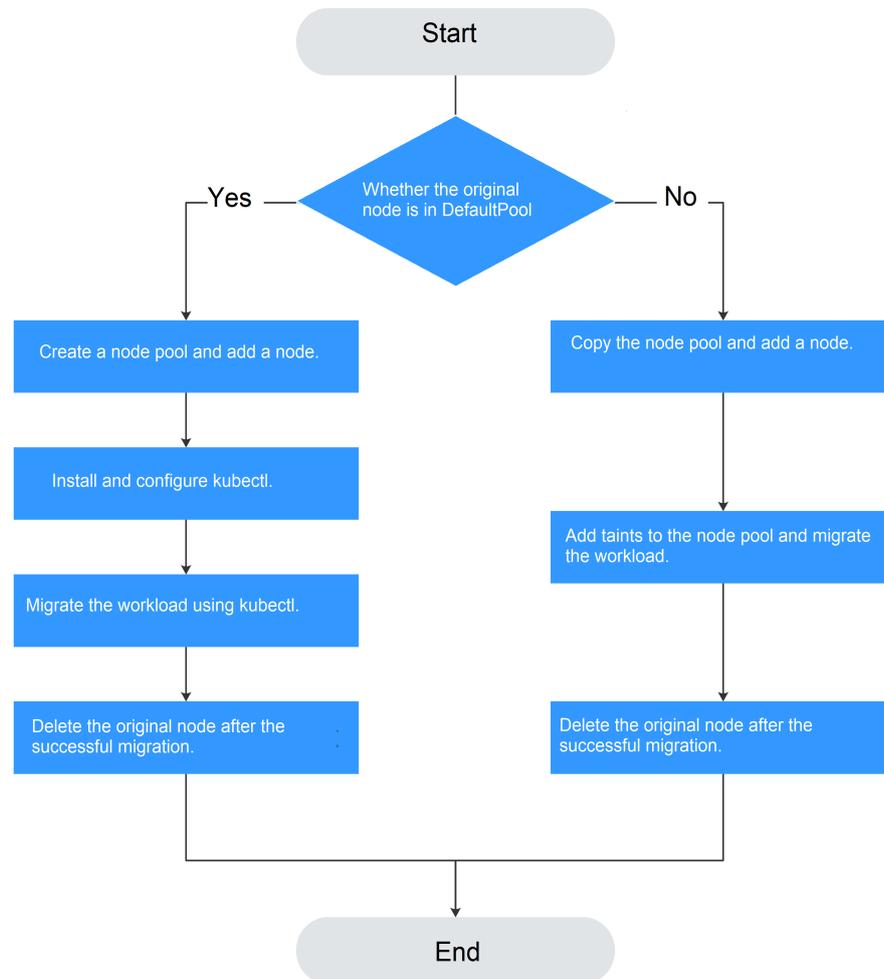
---Fin

## 3.5.9 Realización de actualización de rodamiento para nodos

### Escenario

En una actualización sucesiva, se crea un nuevo nodo, las cargas de trabajo existentes se migran al nuevo nodo y, a continuación, se elimina el nodo antiguo. [Figura 3-8](#) muestra el proceso de migración.

**Figura 3-8** Migración de cargas de trabajo



## Restricciones

- El nodo original y el nodo de destino al que se va a migrar la carga de trabajo deben estar en el mismo clúster.
- El clúster debe ser de v1.13.10 o posterior.
- El grupo de nodos predeterminado DefaultPool no admite esta configuración.

## Escenario 1: El nodo original está en el DefaultPool

- Paso 1** Cree un grupo de nodos. Para obtener más información, véase [Creación de un grupo de nodos](#).
- Paso 2** Haga clic en el nombre del grupo de nodos. La dirección IP del nuevo nodo se muestra en la lista de nodos.
- Paso 3** Instale y configure kubectl. Para obtener más información, véase [Conexión a un clúster con kubectl](#).
- Paso 4** Migre la carga de trabajo.
1. Agregue una mancha al nodo donde la carga de trabajo debe migrarse.  
`kubectl taint node [node] key=value:[effect]`

En el comando anterior, *[node]* indica la dirección IP del nodo donde se encuentra la carga de trabajo que se va a migrar. El valor de *[effect]* puede ser **NoSchedule**, **PreferNoSchedule** o **NoExecute**. En este ejemplo, establezca este parámetro en **NoSchedule**.

- **NoSchedule**: Los pods que no toleran esta mancha no se programan en el nodo; los pods existentes no se desalojan del nodo.
- **PreferNoSchedule**: Kubernetes intenta evitar la programación de pods que no toleran esta mancha en el nodo.
- **NoExecute**: Un pod es desalojado del nodo si ya se está ejecutando en el nodo, y no está programado en el nodo si aún no se está ejecutando en el nodo.

 **NOTA**

Para restablecer una mancha, ejecute el **kubectl taint node [node] key:[effect]- command** para eliminar la mancha.

2. Expulsa de forma segura la carga de trabajo en el nodo.

**kubectl drain [node]**

En el comando anterior, *[node]* indica la dirección IP del nodo donde se encuentra la carga de trabajo que se va a migrar.

3. En el panel de navegación de la consola de CCE, elija **Workloads > Deployments**. En la lista de cargas de trabajo, el estado de la carga de trabajo que se va a migrar cambia de **Running** a **Unready**. Si el estado de la carga de trabajo cambia de nuevo a **Running**, la migración se realiza correctamente.

 **NOTA**

Durante la migración de la carga de trabajo, si se configura la afinidad de nodos para la carga de trabajo, la carga de trabajo sigue mostrando un mensaje que indica que la carga de trabajo no está lista. En este caso, haga clic en el nombre de la carga de trabajo para ir a la página de detalles de la carga de trabajo. En la página de ficha **Scheduling Políticas**, elimine la configuración de afinidad del nodo original y configure las políticas de afinidad y antiafinidad del nuevo nodo. Para obtener más información, véase [Política de programación \(afinidad/antiafinidad\)](#).

Después de migrar correctamente la carga de trabajo, puede ver que la carga de trabajo se migra al nodo creado en **Paso 1** en la página de ficha **Pods** de la página de detalles de la carga de trabajo.

- Paso 5** Elimine el nodo original.

Una vez que la carga de trabajo se haya migrado correctamente y se ejecute correctamente, elimine el nodo original.

----**Fin**

## Escenario 2: El nodo original no está en el DefaultPool

- Paso 1** Copie el grupo de nodos y agréguele nodos. Para obtener más información, véase [Copia de un grupo de nodos](#).

- Paso 2** Haga clic en **View Node** en la columna **Operation** del grupo de nodos. La dirección IP del nuevo nodo se muestra en la lista de nodos.

- Paso 3** Migre la carga de trabajo.

1. Haga clic en **Edit** a la derecha del grupo de nodos original y establezca **Taints**.

2. Introduzca la clave y el valor de la mancha. Las opciones de **Effect** son **NoSchedule**, **PreferNoSchedule** y **NoExecute**. Seleccione **NoExecute** y haga clic en **Add**.
  - **NoSchedule**: Los pods que no toleran esta mancha no se programan en el nodo; los pods existentes no se desalojan del nodo.
  - **PreferNoSchedule**: Kubernetes intenta evitar la programación de pods que no toleran esta mancha en el nodo.
  - **NoExecute**: Un pod es desalojado del nodo si ya se está ejecutando en el nodo, y no está programado en el nodo si aún no se está ejecutando en el nodo.

 **NOTA**

Si necesita restablecer la mancha, elimine el configurado.

3. Haga clic en **OK**.
4. En el panel de navegación de la consola de CCE, elija **Workloads > Deployments**. En la lista de cargas de trabajo, el estado de la carga de trabajo que se va a migrar cambia de **Running** a **Unready**. Si el estado de la carga de trabajo cambia de nuevo a **Running**, la migración se realiza correctamente.

 **NOTA**

Durante la migración de la carga de trabajo, si se configura la afinidad de nodos para la carga de trabajo, la carga de trabajo sigue mostrando un mensaje que indica que la carga de trabajo no está lista. En este caso, haga clic en el nombre de la carga de trabajo para ir a la página de detalles de la carga de trabajo. En la página de ficha **Scheduling Policies**, elimine la configuración de afinidad del nodo original y configure las políticas de afinidad y antiafinidad del nuevo nodo. Para obtener más información, véase [Política de programación \(afinidad/antiafinidad\)](#).

Después de migrar correctamente la carga de trabajo, puede ver que la carga de trabajo se migra al nodo creado en **Paso 1** en la página de ficha **Pods** de la página de detalles de la carga de trabajo.

**Paso 4** Elimine el nodo original.

Una vez que la carga de trabajo se haya migrado correctamente y se ejecute correctamente, elimine el nodo original.

---Fin

## 3.6 Optimizing Node System Parameters

### 3.6.1 Lista de parámetros del sistema de nodos que se pueden optimizar

CCE proporciona parámetros de sistema de nodos predeterminados, que pueden causar cuellos de botella en el rendimiento en algunos escenarios. Por lo tanto, puede personalizar y optimizar algunos parámetros del sistema de nodos. [Lista de parámetros del sistema de nodos que se pueden optimizar](#) describe los parámetros del sistema del nodo.

**AVISO**

- La modificación tiene ciertos riesgos. Usted necesita estar familiarizado con los comandos de Linux y el sistema operativo Linux.
- Los parámetros enumerados en **Tabla 3-29** han sido probados y verificados. **No modifique otros parámetros.** De lo contrario, pueden producirse fallos de nodo.
- Los comandos para modificar los parámetros del sistema de nodos solo son válidos cuando se utilizan las imágenes públicas. Los comandos proporcionados en este documento son de referencia solo cuando se utilizan las imágenes privadas.
- Después de reiniciar el nodo, ejecute el comando **sysctl -p** para actualizar el valor del parámetro.

**Tabla 3-29** Lista de parámetros del sistema que se pueden optimizar

Parámetro	Ubicación de parámetro	Descripción	Referencia
kernel.pid_max	/etc/sysctl.conf	Número máximo de ID de proceso en un nodo  Consultando el parámetro: <pre>sysctl kernel.pid_max</pre>	<b>Cambio de los límites de ID de proceso (kernel.pid_max)</b>
RuntimeMaxUse	/etc/systemd/journald.conf	Límite superior de la memoria ocupada por la caché de log de nodo. Si no se establece este parámetro, una gran cantidad de memoria se ocupará después de que el sistema se ejecute durante mucho tiempo.  Consultando el parámetro: <pre>cat /etc/systemd/journald.conf   grep RuntimeMaxUse</pre>	<b>Cambio del RuntimeMaxUse de la memoria utilizada por la caché de log en un nodo</b>
Openfiles	/etc/security/limits.conf	Número máximo de controladores de archivo para un solo proceso en un nodo. Establezca este parámetro en función de los requisitos de servicio.  Consultando el parámetro: <pre>ulimit -n</pre>	<b>Cambio del número máximo de controladores de archivo para un solo proceso en un nodo</b>

Parámetro	Ubicación de parámetro	Descripción	Referencia
(dentro del contenedor de Openfiles) LimitNOFILE LimitNPROC	<ul style="list-style-type: none"> <li>CentOS/ EulerOS: /usr/lib/ systemd/ system/ docker.servic e</li> <li>Ubuntu: /lib/systemd/ system/ docker.servic e</li> </ul>	<p>Número máximo de controladores de archivo para un solo proceso en un contenedor. Establezca este parámetro en función de los requisitos de servicio.</p> <p>Consultando el parámetro:  <pre>cat /proc/`pidof dockerd`/ limits   grep files</pre></p>	<a href="#">Cambio del número máximo de identificadores de archivo para un proceso de contenedor único</a>
file-max	/etc/sysctl.conf	<p>Número máximo de controladores de archivo en el sistema. Establezca este parámetro en función de los requisitos de servicio.</p> <p>Consultando el parámetro:  <pre>sysctl fs.file-max</pre></p>	<a href="#">Cambio del número máximo de controladores de archivo de nivel de sistema en un nodo</a>
nf_conntrack_buckets nf_conntrack_max	/etc/sysctl.conf	<p>Capacidad de la tabla de seguimiento de conexiones. Establezca este parámetro en función de los escenarios de servicio.</p> <p>Uso del bucket (= / [nf_conntrack_count] / [nf_conntrack_buckets]).</p> <p>Ajuste el valor de los bucket para asegurarse de que el uso del bucket sea inferior a 0.7.</p> <p>Consultando el parámetro:  <pre>sysctl net.netfilter.nf_conntrack_ count sysctl net.netfilter.nf_conntrack_ buckets sysctl net.netfilter.nf_conntrack_ max</pre></p>	<a href="#">Modificación de parámetros de núcleo de nodo</a>

Parámetro	Ubicación de parámetro	Descripción	Referencia
net.netfilter.nf_conntrack_tcp_timeout_close	/etc/sysctl.conf	<p>Tiempo de expiración de la entrada de la conexión en el estado de cierre en la tabla de seguimiento de conexiones. Acortando el tiempo de caducidad puede acelerar el reciclaje.</p> <p>Consultando el parámetro:  <pre>sysctl net.netfilter.nf_conntrack_tcp_timeout_close</pre></p>	
net.netfilter.nf_conntrack_tcp_be_liberal	/etc/sysctl.conf	<p>El valor del parámetro es <b>0</b> o <b>1</b>.</p> <ul style="list-style-type: none"> <li>● <b>0</b>: La función está deshabilitada. Todos los paquetes RST que no están en la ventana TCP se marcan como no válidos.</li> <li>● <b>1</b>: La función está habilitada. Solo los paquetes RST que no están en la ventana TCP se marcan como no válidos. En el caso de contenedores, la activación de este parámetro puede evitar que se limite el ancho de banda de las conexiones TCP que se han traducido mediante NAT.</li> </ul> <p>Consultando el parámetro:  <pre>sysctl net.netfilter.nf_conntrack_tcp_be_liberal</pre></p>	
tcp_keepalive_time	/etc/sysctl.conf	<p>Intervalo en el que se envía un mensaje de mantenimiento de TCP. Si este parámetro se establece en un valor grande, las conexiones TCP pueden suspenderse en la fase <b>Close_wait</b> durante mucho tiempo, agotando los recursos del sistema.</p> <p>Consultando el parámetro:  <pre>sysctl net.ipv4.tcp_keepalive_time</pre></p>	

Parámetro	Ubicación de parámetro	Descripción	Referencia
tcp_max_syn_backlog	/etc/sysctl.conf	Número máximo de medias conexiones de TCP, es decir, el número máximo de conexiones en la cola <b>SYN_RECV</b> .  Consultando el parámetro: <pre>sysctl net.ipv4.tcp_max_syn_backlog</pre>	
tcp_max_tw_buckets	/etc/sysctl.conf	Especifica el número máximo de sockets en el estado <b>time-wait</b> que puede existir en cualquier momento. Si el valor del parámetro es demasiado grande, los recursos de nodo pueden agotarse.  Consultando el parámetro: <pre>sysctl net.ipv4.tcp_max_tw_buckets</pre>	
net.core.somaxconn	/etc/sysctl.conf	Número máximo de conexiones de TCP y tamaño máximo de la cola <b>ESTABLISHED</b> . Si el valor del parámetro es demasiado pequeño, el valor puede ser insuficiente.  Consultando el parámetro: <pre>sysctl net.core.somaxconn</pre>	
max_user_instances	/etc/sysctl.conf	Número máximo de instancias de inotificación permitidas para cada usuario. Si el valor del parámetro es demasiado pequeño, el número de instancias de inotify puede ser insuficiente en los contenedores.  Consultando el parámetro: <pre>sysctl fs.inotify.max_user_instances</pre>	

Parámetro	Ubicación de parámetro	Descripción	Referencia
max_user_watches	/etc/sysctl.conf	Número máximo de directorios de todas las instancias de supervisión. Si el valor del parámetro es demasiado pequeño, el número de directorios puede ser insuficiente en escenarios de contenedor.  Consultando el parámetro: sysctl fs.inotify.max_user_watches	
netdev_max_backlog	/etc/sysctl.conf	Tamaño de la cola de recepción de paquetes de la pila de protocolos de red. Si el valor del parámetro es demasiado pequeño, el tamaño de la cola puede ser insuficiente.  Consultando el parámetro: sysctl net.core.netdev_max_backlog	
net.core.wmem_max net.core.rmem_max	/etc/sysctl.conf	Tamaño de memoria (bytes) del búfer de envío y recepción. Si este parámetro se establece en un valor pequeño, el tamaño de memoria puede ser insuficiente en escenarios de los archivos grandes.  Consultando el parámetro: sysctl net.core.wmem_max sysctl net.core.rmem_max	
net.ipv4.neigh.default.gc_thresh1 net.ipv4.neigh.default.gc_thresh2 net.ipv4.neigh.default.gc_thresh3	/etc/sysctl.conf	Optimización de la recolección de basura de entradas de ARP.  Consultando el parámetro: sysctl net.ipv4.neigh.default.gc_thresh1 sysctl net.ipv4.neigh.default.gc_thresh2 sysctl net.ipv4.neigh.default.gc_thresh3	
vm.max_map_count	/etc/sysctl.conf	Si este parámetro se establece en un valor pequeño, se muestra un mensaje que indica que el espacio es insuficiente durante la instalación de ELK.  Consultando el parámetro: sysctl vm.max_map_count	

## 3.6.2 Cambio del RuntimeMaxUse de la memoria utilizada por la caché de log en un nodo

Journald es un sistema de log en Linux. Escribe información de log en archivos binarios y utiliza el directorio `/run/log/journal` como el directorio de caché de log de forma predeterminada. El archivo de configuración de Journald se almacena en el directorio `/etc/systemd/journald.conf` del nodo. El parámetro **RuntimeMaxUse** indica el uso máximo de memoria de la caché de log. Si **RuntimeMaxUse** no está configurado, una gran cantidad de memoria se ocupará después de que el sistema funcione durante mucho tiempo.

### AVISO

Los comandos para modificar los parámetros del sistema de nodos solo son válidos cuando se utilizan las imágenes públicas. Los comandos proporcionados en este documento son de referencia solo cuando se utilizan las imágenes privadas.

### Cambio de RuntimeMaxUse

**Paso 1** Inicie sesión en el nodo y vea el archivo `/etc/systemd/journald.conf`.

```
cat /etc/systemd/journald.conf
```

**Paso 2** Modifique **RuntimeMaxUse**. El valor recomendado es **100M**.

- Si se ha establecido **RuntimeMaxUse** en el archivo `journald.conf`, ejecute el siguiente comando para cambiar el valor:

```
sed -i "s/RuntimeMaxUse=[0-9]*M/RuntimeMaxUse=100M/g" /etc/systemd/journald.conf && systemctl restart systemd-journald
```

- Si **RuntimeMaxUse** no está definido en el archivo `journald.conf`, ejecute el siguiente comando para agregarlo:

```
echo RuntimeMaxUse=100M >> /etc/systemd/journald.conf && systemctl restart systemd-journald
```

**Paso 3** Si el valor devuelto es el mismo que el valor modificado, la modificación es correcta.

```
cat /etc/systemd/journald.conf | grep RuntimeMaxUse
```

----Fin

### Configuración automática de RuntimeMaxUse al crear un nodo o grupo de nodos

Puede configurar el script para que se ejecute después de que se instale un nodo o un grupo de nodos. Al crear un nodo o grupo de nodos, puede usar el script para configurar el tamaño **RuntimeMaxUse**.

**Paso 1** Confirme el sistema operativo del nodo o grupo de nodos que se va a crear, por ejemplo, CentOS 7.6.

**Paso 2** Pruebe manualmente los comandos de script en los nodos **en el mismo clúster y ejecute el mismo SO**. Para obtener más información sobre cómo ejecutar manualmente el script, consulte [Cambio de RuntimeMaxUse](#).

**Paso 3** Al crear un nodo o grupo de nodos, elija **Advanced Settings > Post-installation Command** para agregar comandos. (Los siguientes comandos deben configurarse después de que la verificación se realice correctamente.)

- Inicie sesión en el nodo y compruebe el archivo `/etc/systemd/journald.conf`. Si se ha definido `RuntimeMaxUse`, ejecute el siguiente comando para cambiar el valor:  

```
sed -i "s/RuntimeMaxUse=[0-9]*M/RuntimeMaxUse=100M/g" /etc/systemd/journald.conf && systemctl restart systemd-journald
```
- Inicie sesión en el nodo y compruebe el archivo `/etc/systemd/journald.conf`. Si `RuntimeMaxUse` no está definido, ejecute el siguiente comando para agregarlo:  

```
echo RuntimeMaxUse=100M >> /etc/systemd/journald.conf && systemctl restart systemd-journald
```

El comando de la siguiente figura solo se utiliza como ejemplo. Cámbielo según sea necesario.

The screenshot shows a configuration interface for ECS Groups. It has three main sections:

- ECS Group:** A dropdown menu currently set to "Anti-affinity" with a help icon. Below it is a selection box with "--Select--" and a "Add ECS Group" button.
- Pre-installation Command:** A text area containing a warning: "Command executed before Kubernetes software is installed. Executing this command may cause the installation to fail. It is commonly used to format data disks." The character count is 0/1,000.
- Post-installation Command:** A text area containing the command: "echo RuntimeMaxUse=100M >> /etc/systemd/journald.conf && systemctl restart systemd-journald". The character count is 91/1,000.

**Paso 4** Una vez creado el nodo, inicie sesión en el nodo para comprobar si la modificación se ha realizado correctamente.

```
cat /etc/systemd/journald.conf | grep RuntimeMaxUse
```

----Fin

### 3.6.3 Cambio del número máximo de controladores de archivo

El número máximo de controladores de archivo es el número máximo de archivos que se pueden abrir. En Linux, hay dos restricciones de manejo de archivos: restricción a nivel de sistema, es decir, el número máximo de archivos que pueden ser abiertos por todos los procesos de usuario al mismo tiempo; restricción a nivel de usuario, es decir, el número máximo de archivos que pueden ser abiertos por un solo proceso de usuario. Los contenedores tienen la tercera restricción de manejadores de archivo, es decir, el número máximo de manejadores de archivo de un solo proceso en el contenedor.

#### AVISO

Los comandos para modificar los parámetros del sistema de nodos solo son válidos cuando se utilizan las imágenes públicas. Los comandos proporcionados en este documento son de referencia solo cuando se utilizan las imágenes privadas.

## Cambio del número máximo de controladores de archivo de nivel de sistema en un nodo

**Paso 1** Inicie sesión en el nodo y vea el archivo `/etc/sysctl.conf`.

```
cat /etc/sysctl.conf
```

**Paso 2** Modifique el parámetro `fs.file-max`. `fs.file-max=1048576` indica el nombre del parámetro del núcleo y el valor recomendado.

- Si el valor de `fs.file-max` se ha establecido en el archivo `sysctl.conf`, ejecute el siguiente comando para cambiar el valor:

```
sed -i "s/fs.file-max=[0-9]*$/fs.file-max=1048576/g" /etc/sysctl.conf &&  
sysctl -p
```

- Si `fs.file-max` no está definido en el archivo `sysctl.conf`, ejecute el siguiente comando para agregarlo:

```
echo fs.file-max=1048576 >> /etc/sysctl.conf && sysctl -p
```

**Paso 3** Ejecute los siguientes comandos para comprobar si el cambio es correcto (si el valor devuelto es el mismo que el configurado).

```
# sysctl fs.file-max  
fs.file-max = 1048576
```

----Fin

## Cambio del número máximo de controladores de archivo para un solo proceso en un nodo

**Paso 1** Inicie sesión en el nodo y vea el archivo `/etc/security/limits.conf`.

```
cat /etc/security/limits.conf
```

El número máximo de controladores de archivo de un solo proceso en un nodo se especifica mediante los siguientes parámetros:

```
...  
root soft nofile 65535  
root hard nofile 65535  
* soft nofile 65535  
* hard nofile 65535
```

**Paso 2** Ejecute el comando `sed` para cambiar el número máximo de controladores de archivo. En el comando `65535` es el número máximo recomendado de manejadores de archivo. El archivo `/etc/security/limits.conf` del nodo de EulerOS 2.3 no contiene la configuración predeterminada relacionada con `nofile`. Por lo tanto, no puede ejecutar el comando `sed` para modificar la configuración.

```
sed -i "s/nofile.[0-9]*$/nofile 65535/g" /etc/security/limits.conf
```

**Paso 3** Inicie sesión de nuevo en el nodo y ejecute el siguiente comando para comprobar si la modificación se realiza correctamente. Si el valor devuelto es el mismo que el valor modificado, la modificación se realiza correctamente.

```
# ulimit -n  
65535
```

----Fin

## Cambio del número máximo de identificadores de archivo para un proceso de contenedor único

**Paso 1** Inicie sesión en el nodo y vea el archivo `/usr/lib/systemd/system/docker.service`.

- CentOS/EulerOS:  

```
cat /usr/lib/systemd/system/docker.service
```
- Ubuntu:  

```
cat /lib/systemd/system/docker.service
```

### NOTA

Si `LimitNOFILE` o `LimitNPROC` se establece en `infinity`, el número máximo de manejadores de archivo soportados por un solo proceso de un contenedor es de **1,048,576**.

El número máximo de controladores de archivo para un solo proceso de un contenedor se especifica mediante los siguientes parámetros:

```
...
LimitNOFILE=1048576
LimitNPROC=1048576
...
```

**Paso 2** Ejecute los siguientes comandos para modificar los dos parámetros. En el comando **1048576** es el valor recomendado del número máximo de manejadores de archivo.

### AVISO

Al cambiar el número máximo de controladores de archivo de un contenedor se reiniciará el proceso Docker.

- CentOS/EulerOS:  

```
sed -i "s/LimitNOFILE=[0-9a-Z]*$/LimitNOFILE=1048576/g" /usr/lib/systemd/system/docker.service;sed -i "s/LimitNPROC=[0-9a-Z]*$/LimitNPROC=1048576/g" /usr/lib/systemd/system/docker.service && systemctl daemon-reload && systemctl restart docker
```
- Ubuntu:  

```
sed -i "s/LimitNOFILE=[0-9a-Z]*$/LimitNOFILE=1048576/g" /lib/systemd/system/docker.service;sed -i "s/LimitNPROC=[0-9a-Z]*$/LimitNPROC=1048576/g" /lib/systemd/system/docker.service && systemctl daemon-reload && systemctl restart docker
```

**Paso 3** Check the maximum number of file handles of a single process in the container. Si el valor devuelto es el mismo que el valor modificado, la modificación se realiza correctamente.

```
# cat /proc/`pidof dockerd`/limits | grep files
Max open files          1048576                1048576                files
```

----Fin

## Configuración automática del número máximo de controladores de archivo al crear un nodo o un grupo de nodos

Puede configurar el script para que se ejecute después de crear un nodo o un grupo de nodos. Al crear un nodo o grupo de nodos, puede utilizar el script para configurar el número máximo de controladores de archivo.

**Paso 1** Confirme el sistema operativo del nodo o grupo de nodos que se va a crear, por ejemplo, CentOS 7.6.

**Paso 2** Pruebe manualmente los comandos de script en los nodos **en el mismo clúster y ejecute el mismo SO.**

- **Cambio del número máximo de controladores de archivo de nivel de sistema en un nodo**
- **Cambio del número máximo de controladores de archivo para un solo proceso en un nodo**
- **Cambio del número máximo de identificadores de archivo para un proceso de contenedor único**

**Paso 3** Al crear un nodo o grupo de nodos, elija **Advanced Settings > Post-installation Command** para agregar comandos. **(Los siguientes comandos deben configurarse después de que la verificación se realice correctamente.)**

- Cambie el número máximo de controladores de archivo de nivel de sistema en un nodo.

- Inicie sesión en el nodo y compruebe el archivo **/etc/sysctl.conf**. Si el valor de **fs.file-max** se ha establecido en el archivo, ejecute el siguiente comando para cambiarlo:

```
sed -i "s/fs.file-max=[0-9]*$/fs.file-max=1048576/g" /etc/sysctl.conf && sysctl -p
```

- Inicie sesión en el nodo y compruebe el archivo **/etc/sysctl.conf**. Si el valor de **fs.file-max** no está establecido en el archivo, ejecute el siguiente comando para agregarlo:

```
echo fs.file-max=1048576 >> /etc/sysctl.conf && sysctl -p
```

En el comando anterior **fs.file-max=1048576** indica el nombre del parámetro del núcleo y el valor recomendado.

- Ejecute el siguiente comando para cambiar el número máximo de controladores de archivo para un solo proceso en un nodo:

```
sed -i "s/nofile.[0-9]*$/nofile 65535/g" /etc/security/limits.conf
```

En el comando anterior, 65535 es el número máximo recomendado de controladores de archivo.

- Cambie el número máximo de controladores de archivo para un solo proceso de un contenedor.

- CentOS/EulerOS:

```
sed -i "s/LimitNOFILE=[0-9a-Z]*$/LimitNOFILE=1048576/g" /usr/lib/systemd/system/docker.service;sed -i "s/LimitNPROC=[0-9a-Z]*$/LimitNPROC=1048576/g" /usr/lib/systemd/system/docker.service && systemctl daemon-reload && systemctl restart docker
```

- Ubuntu:

```
sed -i "s/LimitNOFILE=[0-9a-Z]*$/LimitNOFILE=1048576/g" /lib/systemd/system/docker.service;sed -i "s/LimitNPROC=[0-9a-Z]*$/LimitNPROC=1048576/g" /lib/systemd/system/docker.service && systemctl daemon-reload && systemctl restart docker
```

En el comando anterior, el número máximo de controladores de archivo recomendado es **1048576**.

El comando de la siguiente figura solo se utiliza como ejemplo. Cámbielo según sea necesario.

ECS Group Anti-affinity ?

--Select-- ↕ [Add ECS Group](#)

Pre-installation Command

Command executed before Kubernetes software is installed. Executing this command may cause the installation to fail. It is commonly used to format data disks.

0/1,000

Post-installation Command

echo fs.file-max=1048576 >> /etc/sysctl.conf && sysctl -p

57/1,000

Agency

--Select-- ↕ [Create Agency](#) ?

**Paso 4** Una vez creado el nodo, inicie sesión en el nodo para comprobar si los parámetros se han modificado correctamente.

----Fin

### 3.6.4 Modificación de parámetros de núcleo de nodo

Es posible que los parámetros predeterminados del kernel de Linux no satisfagan a todos los usuarios. Puede modificar el archivo de configuración `/etc/sysctl.conf` en el nodo para modificar los parámetros del núcleo.

#### AVISO

- Los comandos para modificar los parámetros del sistema de nodos solo son válidos cuando se utilizan las imágenes públicas. Los comandos proporcionados en este documento son de referencia solo cuando se utilizan las imágenes privadas.
- Después de reiniciar el nodo, ejecute el comando `sysctl -p` para actualizar el valor del parámetro.

**Tabla 3-30** Parámetros de núcleo de un nodo

Parámetro	Ubicación de parámetro	Descripción	Valor recomendado
file-max	/etc/sysctl.conf	Número máximo de manejadores de archivo en el sistema, que se pueden ajustar según sea necesario.  Consultando el parámetro: <code>sysctl fs.file-max</code>	fs.file-max=1048576

Parámetro	Ubicación de parámetro	Descripción	Valor recomendado
nf_conntrack_buckets nf_conntrack_max	/etc/sysctl.conf	Capacidad de la tabla de seguimiento de conexiones, que se puede ajustar según sea necesario. Uso del bucket = $\frac{[nf\_conntrack\_count]}{[nf\_conntrack\_buckets]}$ Si el uso de CPU es mayor que 0.7 durante mucho tiempo, aumente el valor de los bucket para reducir el uso de CPU a menos de 0.7. Consultando el parámetro: <pre>sysctl net.netfilter.nf_conntrack_count sysctl net.netfilter.nf_conntrack_buckets sysctl net.netfilter.nf_conntrack_max</pre> <p><b>NOTA</b>                      Nota:  <b>net.netfilter.nf_conntrack_buckets</b> en EulerOS 2.3, EulerOS 2.5 y CentOS 7.6 no se pueden modificar editando <b>/etc/sysctl.conf</b> y puede modificar los bucket modificando <b>/sys/module/nf_conntrack/parameters/hashsize</b>.</p>	El valor predeterminado se establece en función del tamaño de la memoria del nodo. Si necesita cambiar el valor, consulte la siguiente fórmula: <ul style="list-style-type: none"> <li>● <math>net.netfilter.nf\_conntrack\_buckets = \frac{[nf\_conntrack\_count]}{0.7}</math></li> <li>● <math>net.netfilter.nf\_conntrack\_max = 4 * [nf\_conntrack\_buckets]</math></li> </ul>
net.netfilter.nf_conntrack_tcp_timeout_close	/etc/sysctl.conf	Tiempo de expiración de la entrada de la conexión en el estado de cierre en la tabla de seguimiento de conexiones. Acortando el tiempo de caducidad puede acelerar el reciclaje. Consultando el parámetro: <pre>sysctl net.netfilter.nf_conntrack_tcp_timeout_close</pre>	net.netfilter.nf_conntrack_tcp_timeout_close=3

Parámetro	Ubicación de parámetro	Descripción	Valor recomendado
net.netfilter.nf_conntrack_tcp_be_liberal	/etc/sysctl.conf	<p>El valor del parámetro es <b>0</b> o <b>1</b>.</p> <ul style="list-style-type: none"> <li>● <b>0</b>: La función está deshabilitada. Todos los paquetes RST que no están en la ventana TCP se marcan como no válidos.</li> <li>● <b>1</b>: La función está habilitada. Solo los paquetes RST que no están en la ventana TCP se marcan como no válidos. En el caso de contenedores, la activación de este parámetro puede evitar que se limite el ancho de banda de las conexiones TCP que se han traducido mediante NAT.</li> </ul> <p>Consultando el parámetro:</p> <pre>sysctl net.netfilter.nf_conntrack_tcp_be_liberal</pre>	net.netfilter.nf_conntrack_tcp_be_liberal=1
tcp_keepalive_time	/etc/sysctl.conf	<p>Intervalo para enviar mensajes de detección de Keepalive con TCP. Si este parámetro se establece en un valor grande, las conexiones TCP pueden suspenderse en la fase <b>Close_wait</b> durante mucho tiempo, agotando los recursos del sistema.</p> <p>Consultando el parámetro:</p> <pre>sysctl net.ipv4.tcp_keepalive_time</pre>	net.ipv4.tcp_keepalive_time=600
tcp_max_syn_backlog	/etc/sysctl.conf	<p>Número máximo de medias conexiones de TCP, es decir, el número máximo de conexiones en la cola <b>SYN_RECV</b>.</p> <p>Consultando el parámetro:</p> <pre>sysctl net.ipv4.tcp_max_syn_backlog</pre>	net.ipv4.tcp_max_syn_backlog=8096

Parámetro	Ubicación de parámetro	Descripción	Valor recomendado
tcp_max_tw_buckets	/etc/sysctl.conf	<p>Especifica el número máximo de sockets en el estado de espera de tiempo que puede existir en cualquier momento. Si el valor del parámetro es demasiado grande, los recursos de nodo pueden agotarse.</p> <p>Consultando el parámetro:</p> <pre>sysctl net.ipv4.tcp_max_tw_buckets</pre>	net.ipv4.tcp_max_tw_buckets=5000
net.core.somaxconn	/etc/sysctl.conf	<p>Número máximo de conexiones de TCP y tamaño máximo de la cola ESTABLISHED. Si el valor del parámetro es demasiado pequeño, el valor puede ser insuficiente.</p> <p>Consultando el parámetro:</p> <pre>sysctl net.core.somaxconn</pre>	net.core.somaxconn=32768
max_user_instances	/etc/sysctl.conf	<p>Número máximo de instancias de notificación permitidas para cada usuario. Si el valor del parámetro es demasiado pequeño, el número de instancias de inotify puede ser insuficiente en los contenedores.</p> <p>Consultando el parámetro:</p> <pre>sysctl fs.inotify.max_user_instances</pre>	fs.inotify.max_user_instances=8192
max_user_watches	/etc/sysctl.conf	<p>Número máximo de directorios de todas las instancias de supervisión. Si el valor del parámetro es demasiado pequeño, el número de directorios puede ser insuficiente en los contenedores.</p> <p>Consultando el parámetro:</p> <pre>sysctl fs.inotify.max_user_watches</pre>	fs.inotify.max_user_watches=524288

Parámetro	Ubicación de parámetro	Descripción	Valor recomendado
netdev_max_backlog	/etc/sysctl.conf	Tamaño de la cola de recepción de paquetes de la pila de protocolos de red. Si el valor del parámetro es demasiado pequeño, el tamaño de la cola puede ser insuficiente.  Consultando el parámetro: sysctl net.core.netdev_max_backlog	net.core.netdev_max_backlog=16384
net.core.wmem_max net.core.rmem_max	/etc/sysctl.conf	Tamaño de memoria (bytes) del búfer de envío y recepción. Si este parámetro se establece en un valor pequeño, el tamaño de memoria puede ser insuficiente en escenarios de los archivos grandes.  Consultando el parámetro: sysctl net.core.wmem_max sysctl net.core.rmem_max	net.core.wmem_max=16777216 net.core.rmem_max=16777216
net.ipv4.neigh.default.gc_thresh1 net.ipv4.neigh.default.gc_thresh2 net.ipv4.neigh.default.gc_thresh3	/etc/sysctl.conf	Optimización de la recolección de basura de las entradas ARP.  Consultando el parámetro: sysctl net.ipv4.neigh.default.gc_thresh1 sysctl net.ipv4.neigh.default.gc_thresh2 sysctl net.ipv4.neigh.default.gc_thresh3	net.ipv4.neigh.default.gc_thresh1=0 net.ipv4.neigh.default.gc_thresh2=4096 net.ipv4.neigh.default.gc_thresh3=8192
vm.max_map_count	/etc/sysctl.conf	Si este parámetro se establece en un valor pequeño, se muestra un mensaje que indica que el espacio es insuficiente durante la instalación de ELK.  Consultando el parámetro: sysctl vm.max_map_count	vm.max_map_count=262144

## Modificación de parámetros de núcleo de un nodo

**Tabla 3-30** enumera los parámetros del núcleo de los nodos. A continuación se describe cómo cambiar el valor de **tcp\_keepalive\_time** que indica el intervalo para enviar mensajes de detección de Keepalive a través de TCP.

**Paso 1** Inicie sesión en el nodo y vea el archivo **/etc/sysctl.conf**.

```
cat /etc/sysctl.conf
```

**Paso 2** Modifique el parámetro `net.ipv4.tcp_keepalive_time`. `net.ipv4.tcp_keepalive_time=600` indica el nombre del parámetro del núcleo y el valor recomendado. Para obtener más información sobre el valor recomendado, consulte [Tabla 3-30](#).

Para modificar otros parámetros del núcleo, reemplace los nombres y valores de los parámetros de los comandos haciendo referencia a [Tabla 3-30](#).

- Si se ha establecido `net.ipv4.tcp_keepalive_time` en el archivo `sysctl.conf`, ejecute el siguiente comando para cambiar el valor:

```
sed -i "s/net.ipv4.tcp_keepalive_time=[0-9]*$/  
net.ipv4.tcp_keepalive_time=600/g" /etc/sysctl.conf && sysctl -p
```

- Si `net.ipv4.tcp_keepalive_time` no está definido en el archivo `sysctl.conf`, ejecute el siguiente comando para agregarlo:

```
echo net.ipv4.tcp_keepalive_time=600 >> /etc/sysctl.conf && sysctl -p
```

**Paso 3** Ejecute el comando de [Tabla 3-30](#) para comprobar si la modificación se realiza correctamente. Si el valor devuelto es el mismo que el valor modificado, la modificación se realiza correctamente.

```
# sysctl net.ipv4.tcp_keepalive_time  
net.ipv4.tcp_keepalive_time = 600
```

---Fin

## Configuración automática de parámetros de núcleo al crear un nodo o un grupo de nodos

Puede configurar el script para que se ejecute después de crear un nodo o un grupo de nodos. Al crear un nodo o grupo de nodos, puede usar el script para configurar los parámetros del núcleo.

El parámetro `tcp_keepalive_time` se utiliza como ejemplo para describir cómo cambiar el intervalo para enviar mensajes de detección de Keepalive a través de TCP. El valor es el valor recomendado de [Tabla 3-30](#).

**Paso 1** Confirme el sistema operativo del nodo o grupo de nodos que se va a crear, por ejemplo, CentOS 7.6.

**Paso 2** Pruebe manualmente los comandos de script en los nodos **en el mismo clúster y ejecute el mismo SO**. Para obtener más información sobre cómo ejecutar manualmente el script, consulte [Modificación de parámetros de núcleo de un nodo](#).

**Paso 3** Al crear un nodo o grupo de nodos, elija **Advanced Settings > Post-installation Command** para agregar comandos. (**Los siguientes comandos deben configurarse después de que la verificación se realice correctamente.**) Para modificar otros parámetros del kernel, reemplace los nombres y valores de los parámetros de los comandos haciendo referencia a [Tabla 3-30](#).

- Inicie sesión en el nodo y compruebe el archivo `/etc/sysctl.conf`. Si se ha establecido `net.ipv4.tcp_keepalive_time` en el archivo, ejecute el siguiente comando para cambiarlo:

```
sed -i "s/net.ipv4.tcp_keepalive_time=[0-9]*$/  
net.ipv4.tcp_keepalive_time=600/g" /etc/sysctl.conf && sysctl -p
```

- Inicie sesión en el nodo y compruebe el archivo `/etc/sysctl.conf`. Si `net.ipv4.tcp_keepalive_time` no está definido en el archivo, ejecute el siguiente comando para agregarlo:

```
echo net.ipv4.tcp_keepalive_time=600 >> /etc/sysctl.conf && sysctl -p
```

El comando de la siguiente figura solo se utiliza como ejemplo. Cámbielo según sea necesario.

ECS Group Anti-affinity ?

--Select-- C Add ECS Group [↗](#)

Pre-installation Command 0/1,000

Command executed before Kubernetes software is installed. Executing this command may cause the installation to fail. It is commonly used to format data disks.

Post-installation Command 69/1,000

echo net.ipv4.tcp\_keepalive\_time=600 >> /etc/sysctl.conf &&  
sysctl -p

Agency --Select-- C Create Agency [↗](#) ?

**Paso 4** Una vez creado el nodo, inicie sesión en el nodo y ejecute el comando en [Tabla 3-30](#) para comprobar si la modificación se ha realizado correctamente.

----Fin

## 3.6.5 Cambio de los límites de ID de proceso (kernel.pid\_max)

### Contexto

Los ID de proceso (PID) son un recurso fundamental en los nodos. Es trivial alcanzar el límite de tareas sin alcanzar ningún otro límite de recursos, lo que puede causar inestabilidad a una máquina host.

Puede ajustar el límite de PID (kernel.pid\_max) de acuerdo con los requisitos de servicio.

### kernel.pid\_max predeterminados

A partir de enero de 2022, CCE cambia el valor predeterminado de **kernel.pid\_max** a **4194304** para los nodos de EulerOS 2.5, CentOS 7.6 y Ubuntu 18.04 en clústeres de v1.17 o posterior. Condiciones específicas:

- Versión del clúster: v1.17.17 o posterior
- Creación del nodo: después del 30 de enero de 2022

Si no se cumplen las dos condiciones anteriores, el **kernel.pid\_max** en los nodos de EulerOS 2.5, CentOS 7.6 y Ubuntu 18.04 es **32768** por defecto.

**Tabla 3-31** kernel.pid\_max predeterminados

SO	Clústeres de 1.17.9 y anteriores	Clústeres de 1.17.17 y posteriores	
		Nodos creados el 30 de enero de 2022 o antes	Nodos creados después del 30 de enero de 2022

EulerOS 2.5	32768	32768	4194304
CentOS 7.6	32768	32768	4194304
Ubuntu 18.04	N/A	32768	4194304
EulerOS 2.3	57344	57344	57344
EulerOS 2.9	N/A	4194304	4194304

### Sugerencia de cambio

- EulerOS 2.3: Cambie el valor predeterminado a **4194304** para todos los nodos. Para obtener más información, véase [Cambio de kernel.pid\\_max de un nodo](#). Utilice un script de preinstalación para hacerlo para nuevos nodos y grupos de nodos. Para más detalles, véase [Configuración de kernel.pid\\_max al crear un grupo de nodos](#) y [Configuración de kernel.pid\\_max al crear un nodo](#).
- EulerOS 2.5, CentOS 7.6 y Ubuntu 18.04:
  - Cambie el valor de **kernel.pid\_max** a **4194304** para los nodos creados el 30 de enero de 2022 o anteriores en clústeres de v1.17.17 o posterior. Para obtener más información, véase [Cambio de kernel.pid\\_max de un nodo](#).
  - Para clústeres de 1.17.9 y anteriores:
    - Cambie el valor de **kernel.pid\_max** a **4194304** para los nodos existentes. Para obtener más información, véase [Cambio de kernel.pid\\_max de un nodo](#).
    - Utilice un script de preinstalación para hacerlo para nuevos nodos y grupos de nodos. Para más detalles, véase [Configuración de kernel.pid\\_max al crear un grupo de nodos](#) y [Configuración de kernel.pid\\_max al crear un nodo](#).

## Consulta de kernel.pid\_max

Log in to the node and run the following command to query the value of **kernel.pid\_max**:

```
sysctl kernel.pid_max
```

```
# sysctl kernel.pid_max
kernel.pid_max = 32768
```

Si es necesario, cambie **kernel.pid\_max** según lo indicado en [Cambio de kernel.pid\\_max de un nodo](#).

## Comprobación de PID de nodo

Inicie sesión en el nodo y ejecute el siguiente comando para comprobar cuántos PID están en uso:

```
ps -eflL | wc -l
```

```
# ps -eflL | wc -l
691
```

## Cambio de kernel.pid\_max de un nodo

Inicie sesión en el nodo y ejecute el siguiente comando. **4194304** indica el valor de **kernel.pid\_max** y se usa como ejemplo aquí.

```
echo kernel.pid_max = 4194304 >> /etc/sysctl.conf && sysctl -p
```

```
echo 4194304 > /sys/fs/cgroup/pids/kubepods/pids.max
```

Ejecute los siguientes comandos para comprobar si el cambio es correcto (si el valor devuelto es el mismo que el configurado).

```
# sysctl kernel.pid_max
kernel.pid_max = 4194304
# cat /sys/fs/cgroup/pids/kubepods/pids.max
4194304
```

## Configuración de kernel.pid\_max al crear un grupo de nodos

EulerOS 2.3: Se requiere la configuración.

EulerOS 2.5, CentOS 7.6 y Ubuntu 18.04: **Configuration required** para clústeres de v1.17.9 y anteriores. **Configuration NOT required** para clústeres de v1.17.17 y posteriores porque se ha cambiado el valor.

Puede configurar **kernel.pid\_max** en el script de preinstalación para crear un nodo a partir de un grupo de nodos.

Al crear un grupo de nodos, elija **Advanced Settings > Post-installation Command** y agregue el siguiente comando:

```
echo kernel.pid_max = 4194304 >> /etc/sysctl.conf && sysctl -p
```

## Configuración de kernel.pid\_max al crear un nodo

EulerOS 2.3: Se requiere la configuración.

EulerOS 2.5, CentOS 7.6 y Ubuntu 18.04: **Configuration required** para clústeres de v1.17.9 y anteriores. **Configuration NOT required** para clústeres de v1.17.17 y posteriores porque se ha cambiado el valor.

Puede configurar **kernel.pid\_max** mediante el script de preinstalación al crear un nodo.

Elija **Advanced Settings > Post-installation Command** y agregue el siguiente comando:

```
echo kernel.pid_max = 4194304 >> /etc/sysctl.conf && sysctl -p
```

ECS Group	<div style="border: 1px solid #ccc; padding: 5px;"><span style="background-color: #007bff; color: white; padding: 2px 5px; border-radius: 3px;">Anti-affinity</span> <span style="font-size: 1.2em; color: #007bff;">?</span></div> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;">--Select-- <span style="float: right;">C Add ECS Group <a href="#">↗</a></span></div>
Pre-installation Command	<div style="border: 1px solid #ccc; padding: 5px; min-height: 60px;">Command executed before Kubernetes software is installed. Executing this command may cause the installation to fail. It is commonly used to format data disks.</div>
Post-installation Command	<div style="border: 1px solid #ccc; padding: 5px; min-height: 60px;"><pre>echo kernel.pid_max = 4194304 &gt;&gt; /etc/sysctl.conf &amp;&amp; sysctl -p</pre></div>

## 3.7 Migración de nodos de Docker a containerd

### Contexto

Kubernetes ha eliminado dockershim de v1.24 y no es compatible con Docker de forma predeterminada. CCE seguirá soportando Docker en la v1.25 pero solo hasta la v1.27. Los siguientes pasos muestran cómo migrar nodos de Docker a containerd.

### Requisitos previos

- Se ha creado al menos un clúster que admite nodos de containerd. Para obtener más información, véase [Asignación entre los sistemas operativos de nodos y los motores de contenedores](#).
- Hay un nodo de Docker o un grupo de nodos de Docker en el clúster.

### Precauciones

- Teóricamente, la migración durante la ejecución de contenedor interrumpirá los servicios durante un corto período de tiempo. Por lo tanto, se recomienda encarecidamente que los servicios que se van a migrar se hayan desplegado como varias instancias. Además, se recomienda probar el impacto de la migración en el entorno de prueba para minimizar los riesgos potenciales.
- containerd no puede crear imágenes. No utilice el comando **docker build** para generar imágenes en nodos de containerd. Para otras diferencias entre Docker y containerd, véase [Descripción del motor de contenedores](#).

### Migración de un nodo

- Paso 1** Inicie sesión en la consola de CCE y haga clic en el nombre del clúster para acceder al clúster.
- Paso 2** En el panel de navegación, elija **Nodos**. En la lista de nodos, seleccione uno o más nodos que desea restablecer y elija **More > Reset Node**.
- Paso 3** Ajuste **Container Engine** a **containerd**. Puede ajustar otros parámetros según sea necesario o conservarlos según se establezca durante la creación.

**Compute Settings** Configure the specifications and OS of a cloud server, on which your container

Specifications **General computing-plus | ac7.large.2 | 2cores | 4GiB**

Container Engine **Docker** containerd

OS **Public image** Private image ?

**EulerOS 2.9** CentOS 7.6 **Ubuntu 18.04**

Login Mode **Password** Key Pair

Username **root**

Password

**Paso 4** Si el estado del nodo es **Installing**, se está restableciendo el nodo.

Cuando el estado del nodo es **Running**, puede ver que la versión del nodo cambia a containerd. Puede iniciar sesión en el nodo y ejecutar comandos de containerd como **crictl** para ver información sobre los contenedores que se ejecuta en el nodo.

----Fin

## Migración de un grupo de nodos

Puede **copiar un grupo de nodos**, establecer el motor contenedor del nuevo grupo de nodos en containerd y mantener otras configuraciones iguales a las del grupo de nodos de Docker original.

**Paso 1** Inicie sesión en la consola de CCE y haga clic en el nombre del clúster para acceder al clúster.

**Paso 2** En el panel de navegación, elija **Nodes**. En la página de ficha **Node Pools**, busque el grupo de nodos de Docker que desea copiar y elija **More > Copy** en la columna **Operation**.

Node Pool Name	Status	Actual/Desired Nodes	Specifications	AZ	Billing Mode	Auto Scaling	Operation
y00576872-emi-12170-nodepool-66678	Normal	0/0	Elastic Cloud Server (VM)   c7.star...	Random	Pay-per-use	Close	View Node Edit More <b>Copy</b> Delete

**Paso 3** En el área **Compute Settings**, establezca **Container Engine** en **containerd** y modifique otros parámetros según sea necesario.

**Compute Settings** Configure the specifications and OS of a cloud server, on which your containerized applications run.

Billing Mode **Pay-per-use** Yearly/Monthly ?

AZ **Random** AZ1 AZ2 AZ3

AZ where the node is located. Nodes in a cluster can be created in different AZs for higher reliability **Not editable after creation**

Node Type **Elastic Cloud Server (VM)** Elastic Cloud Server (physical machine) BMS ?

Container Engine **containerd** Docker

Specifications vCPUs  Memory  Flavor

**Paso 4** Escale el número de grupos de nodos en containerd creados al número de grupos de nodos de Docker originales y eliminar los nodos de los grupos de nodos de Docker uno por uno.

Se prefiere la migración de balanceo. Es decir, agregue algunos nodos en containerd y luego elimine algunos nodos de Docker hasta que el número de nodos en el nuevo grupo de nodos en containerd sea el mismo que en el grupo de nodos Docker original.

 **NOTA**

Si ha establecido afinidad de nodos para las cargas de trabajo desplegadas en los nodos o el grupo de nodos de Docker originales, debe establecer políticas de afinidad para que las cargas de trabajo se ejecuten en los nuevos nodos o grupos de nodos containerd.

**Paso 5** Después de la migración, elimine el grupo de nodos de Docker original.

----**Fin**

# 4 Grupos de nodos

---

## 4.1 Descripción de pool de nodos

### Presentación

CCE presenta los grupos de nodos para ayudarle a gestionar mejor los nodos en los clústeres de Kubernetes. Un grupo de nodos contiene un nodo o un grupo de nodos con una configuración idéntica en un clúster.

Puede crear grupos de nodos personalizados en la consola de CCE. Con los grupos de nodos, puede crear, gestionar y destruir rápidamente nodos sin afectar al clúster. Todos los nodos de un grupo de nodos personalizado tienen parámetros y tipo de nodo idénticos. No se puede configurar un solo nodo en un grupo de nodos; cualquier cambio de configuración afecta a todos los nodos del grupo de nodos.

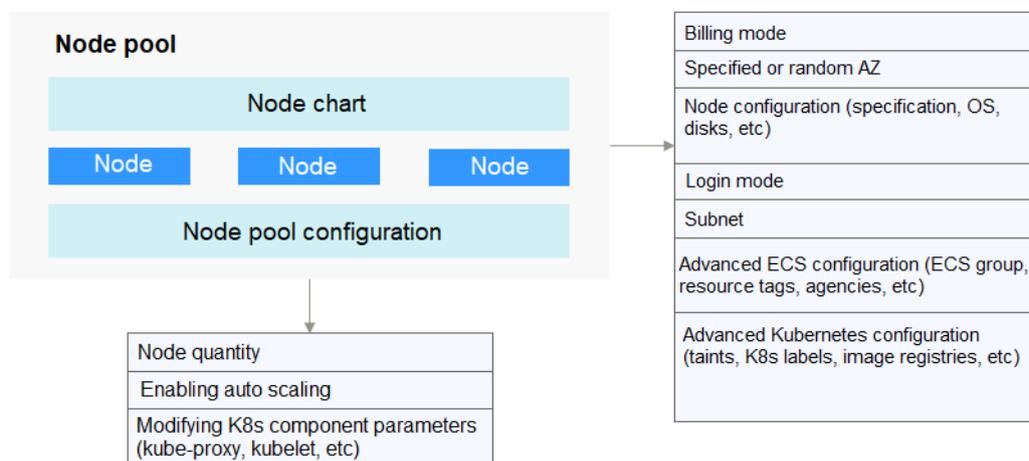
También puede utilizar grupos de nodos para el ajuste automático (soportado únicamente por grupos de nodos de pago por uso).

- Cuando un pod de un clúster no se puede programar debido a la insuficiencia de recursos, se pueden activar automáticamente la expansión horizontal.
- Cuando hay un nodo inactivo o se cumple un umbral de métrica de monitorización, la reducción horizontal se puede activar automáticamente.

Esta sección describe cómo funcionan los grupos de nodos en CCE y cómo crear y gestionar grupos de nodos.

## Arquitectura de grupo de nodos

Figura 4-1 Arquitectura general de un grupo de nodos



En general, todos los nodos de un grupo de nodos tienen los mismos atributos:

- SO del nodo
- Especificaciones del nodo.
- Modo de inicio de sesión de nodo.
- Tiempo de ejecución del nodo.
- Parámetros de inicio de los componentes de Kubernetes en un nodo
- Script de inicio definido por el usuario de un nodo
- **Kubernetes Labels y Taints**

CCE proporciona los siguientes atributos extendidos para grupos de nodos:

- Sistema operativo del grupo de nodos
- Número máximo de pods en cada nodo de un grupo de nodos

## Descripción del DefaultPool

DefaultPool no es un grupo de nodos real. Solo **clasifica** los nodos que no están en los grupos de nodos creados por el usuario. Estos nodos se crean directamente en la consola o invocando a las API. DefaultPool no admite ninguna función de grupo de nodos creada por el usuario, incluida la configuración de ajuste y parámetros. DefaultPool no se puede editar, eliminar, expandir o escalar automáticamente, y los nodos de él no se pueden migrar.

## Escenarios posibles

Cuando se requiere un clúster a gran escala, se recomienda utilizar grupos de nodos para gestionar los nodos.

En la siguiente tabla se describen varios escenarios de gestión de clústeres a gran escala y las funciones de grupos de nodos en cada escenario.

**Tabla 4-1** Uso de grupos de nodos para diferentes escenarios de gestión

Escenario	Función
Múltiples nodos heterogéneos (con diferentes modelos y configuraciones) en el clúster	Los nodos se pueden agrupar en diferentes grupos para la gestión.
Se requiere ajuste de nodos frecuente en un clúster	Los grupos de nodos admiten el ajuste automático para agregar o reducir nodos de forma dinámica.
Reglas de programación de aplicaciones complejas en un clúster	Las etiquetas de grupo de nodos se pueden utilizar para especificar rápidamente reglas de programación de servicios.

## Funciones y precauciones

Función	Descripción	Notas
Creación de un grupo de nodos	Agregar un grupo de nodos.	Se recomienda que un clúster no contenga más de 100 grupos de nodos.
Eliminación de un grupo de nodos	Cuando se elimina un grupo de nodos, los nodos del grupo de nodos se eliminan primero. Cuando se elimina un grupo de nodos anual/mensual, los nodos se migrarán primero al grupo de nodos predeterminado. Las cargas de trabajo de los nodos originales se migran automáticamente a los nodos disponibles en otros grupos de nodos.	Si los pods en el grupo de nodos tienen un selector de nodos específico y ninguno de los otros nodos en el clúster satisface el selector de nodos, los pods se volverán no programables.
Habilitar el ajuste automático para un grupo de nodos	Después de activar el ajuste automático, los nodos se crearán o eliminarán automáticamente en el grupo de nodos en función de las cargas del clúster.	Se recomienda no almacenar datos importantes en los nodos de un grupo de nodos porque después del ajuste automático, los datos no se pueden restaurar ya que los nodos se pueden eliminar.
Habilitar el ajuste automático para un grupo de nodos	Después de deshabilitar el ajuste automático, el número de nodos en un grupo de nodos no cambiará automáticamente con las cargas del clúster.	/

Función	Descripción	Notas
Ajustar el tamaño de un grupo de nodos	El número de nodos en un grupo de nodos se puede ajustar directamente. Si se reduce el número de nodos, los nodos se eliminan aleatoriamente del grupo de nodos actual.	Después de activar el ajuste automático, no se recomienda ajustar manualmente el tamaño del grupo de nodos.
Cambiar las configuraciones de grupo de nodos	Puede modificar el nombre del grupo de nodos, la cantidad de nodos, las etiquetas de Kubernetes (y su cantidad), las etiquetas de recursos y las manchas y ajustar la configuración del disco, el SO y el motor de contenedor del grupo de nodos.	Las etiquetas y manchas de Kubernetes eliminadas o agregadas (así como su cantidad) se aplicarán a todos los nodos del grupo de nodos, lo que puede provocar una reprogramación de pods. Por lo tanto, realice esta operación con precaución.
Eliminación de un nodo de un grupo de nodos	Los nodos de un grupo de nodos se pueden migrar al grupo de nodos predeterminado del mismo clúster.	Los nodos del grupo de nodos predeterminado no se pueden migrar a otros grupos de nodos y los nodos de un grupo de nodos creado por el usuario no se pueden migrar a otros grupos de nodos creados por el usuario.
Clonar un grupo de nodos	Puede copiar la configuración de un grupo de nodos existente para crear un nuevo grupo de nodos.	/
Ajustar los parámetros de Kubernetes	Puede configurar los componentes principales con granularidad fina.	<ul style="list-style-type: none"> <li>● Esta función solo se admite en clústeres de v1.15 y posteriores. No se muestra para las versiones anteriores a la v1.15.</li> <li>● El grupo de nodos predeterminado DefaultPool no admite este tipo de configuración.</li> </ul>

## Despliegue de una carga de trabajo en un grupo de nodos especificado

Al crear una carga de trabajo, puede restringir que los pods se ejecuten en un grupo de nodos especificado.

Por ejemplo, en la consola de CCE, puede establecer la afinidad entre la carga de trabajo y el nodo en la página de ficha **Scheduling Policies** de la página de detalles de la carga de trabajo para desplegar a la fuerza la carga de trabajo en un grupo de nodos específico. De esta manera, la carga de trabajo se ejecuta solo en los nodos del grupo de nodos. Si necesita controlar mejor dónde se programará la carga de trabajo, puede utilizar políticas de afinidad o antiafinidad entre las cargas de trabajo y los nodos que se describen en [Política de programación \(afinidad/antiafinidad\)](#).

Por ejemplo, puede usar la solicitud de recurso de contenedor como un `nodeSelector` para que las cargas de trabajo se ejecuten solo en los nodos que cumplan con la solicitud de recurso.

Si el archivo de definición de carga de trabajo define un contenedor que requiere cuatro CPU, el planificador no elegirá los nodos con dos CPU para ejecutar las cargas de trabajo.

## Operaciones relacionadas

Puede iniciar sesión en la consola de CCE y consultar las siguientes secciones para realizar operaciones en grupos de nodos:

- [Creación de un grupo de nodos](#)
- [Gestión de un grupo de nodos](#)
- [Creación de una Deployment](#)
- [Política de programación \(afinidad/antiafinidad\)](#)

## 4.2 Creación de un grupo de nodos

### Escenario

Esta sección describe cómo crear un grupo de nodos y realizar operaciones en el grupo de nodos. Para obtener más información sobre el funcionamiento de un grupo de nodos, consulte [Descripción de pool de nodos](#).

### Notas y restricciones

- El complemento del autoscaler debe instalarse para el ajuste automático de nodos. Para obtener más información acerca de la instalación del complemento y la configuración de parámetros, consulte [autoscaler](#).
- El ajuste automático solo está disponible para grupos de nodos de pago por uso, no para aquellos que se facturan por año o mes.
- Solo los clústeres de v1.19 o posterior admiten grupos de seguridad personalizados.

### Procedimiento

**Paso 1** Inicie sesión en la consola de CCE.

**Paso 2** Haga clic en el nombre del clúster y acceda a la consola del clúster. Elija **Nodos** en el panel de navegación de la izquierda y haga clic en la ficha **Node Pools** de la derecha.

**Paso 3** En la esquina superior derecha de la página, haga clic en **Create Node Pool**.

#### Ajustes básicos

**Tabla 4-2** Configuración básica

Parámetro	Descripción
Node Pool Name	Nombre de un grupo de nodos. De forma predeterminada, el nombre tiene el formato de <i>Cluster name-nodepool-Random number</i> . Si no desea utilizar el formato de nombre predeterminado, puede personalizar el nombre.

Parámetro	Descripción
Nodes	Número de nodos que se van a crear en este grupo de nodos.

Parámetro	Descripción
Auto Scaling	<p>De forma predeterminada, el ajuste automático está deshabilitado. El ajuste automático solo está disponible para grupos de nodos de pago por uso, no para aquellos que se facturan por año o mes. Para habilitar el ajuste automático, instale el complemento del <a href="#">autoscaler</a>.</p> <p>Después de habilitar el ajuste automático activando , los nodos del grupo de nodos se crearán o eliminarán automáticamente en función de las cargas del clúster.</p> <ul style="list-style-type: none"> <li>● <b>Maximum Nodes y Minimum Nodes:</b> Puede establecer el número máximo y mínimo de nodos para asegurarse de que el número de nodos a escalar está dentro de un rango adecuado.</li> <li>● <b>Priority:</b> Establezca este parámetro en función de los requisitos de servicio. Un valor mayor indica una prioridad más alta. Por ejemplo, si este parámetro se establece en <b>1</b> y <b>4</b> respectivamente para los grupos de nodos A y B, B tiene una prioridad más alta que A. Si las prioridades de múltiples grupos de nodos se establecen en el mismo valor, por ejemplo, el <b>2</b> los grupos de nodos no se priorizan y el sistema realiza el ajuste basándose en el principio mínimo de desperdicio de recursos.</li> </ul> <p><b>NOTA</b>                      CCE selecciona un grupo de nodos para escalar automáticamente según las siguientes políticas:</p> <ol style="list-style-type: none"> <li>1. CCE usa algoritmos para determinar si un grupo de nodos cumple las condiciones para permitir la planificación de un pod en estado pendiente, incluido si los recursos de nodo son mayores que los solicitados por el pod, y si el nodeSelect, nodeAffinity y manchas cumplen las condiciones. Además, se filtran los grupos de nodos que no se pueden escalar (debido a recursos insuficientes u otras razones) y que todavía están en el intervalo de enfriamiento de 15 minutos.</li> <li>2. Si varios grupos de nodos cumplen con los requisitos de ajuste, el sistema comprueba la prioridad de cada grupo de nodos y selecciona el grupo de nodos con la prioridad más alta para escalar. El valor varía de 0 a 100 y la prioridad predeterminada es 0. El valor 100 indica la prioridad más alta, y el valor 0 indica la prioridad más baja.</li> <li>3. Si varios grupos de nodos tienen la misma prioridad o no se ha configurado ninguna prioridad para ellos, el sistema selecciona el grupo de nodos que consumirá el menor número de recursos según la especificación de VM configurada.</li> <li>4. Si las especificaciones de VM de múltiples grupos de nodos son las mismas pero los grupos de nodos se despliegan en diferentes AZ, el sistema selecciona aleatoriamente un grupo de nodos para activar el ajuste.</li> </ol> <ul style="list-style-type: none"> <li>● <b>Cooldown Period:</b> Ingrese un período, en minutos. Este campo indica el período durante el cual los nodos agregados en el grupo de nodos actual no se pueden escalar. Los intervalos de enfriamiento de escalado se pueden configurar en la configuración del grupo de nodos y en la configuración del <a href="#">complemento del autoscaler</a>.</li> </ul> <p><b>Intervalo de enfriamiento de reducción configurado en un grupo de nodos</b></p>

Parámetro	Descripción
	<p>Este intervalo indica el período durante el cual no se pueden eliminar los nodos agregados al grupo de nodos actual después de una operación de expansión. Este intervalo tiene efecto en el nivel del grupo de nodos.</p> <p><b>Intervalo de enfriamiento de reducción configurado en el complemento del autoscaler</b></p> <p>El intervalo después de una expansión indica el período durante el cual no se puede escalar todo el clúster después de que el complemento de autoscaler active expansión (debido a los pods, las métricas y las políticas de ajuste no programables). Este intervalo tiene efecto a nivel de clúster.</p> <p>El intervalo después de eliminar un nodo indica el período durante el cual no se puede escalar el clúster después de que el complemento de autoscaler active reducción. Este intervalo tiene efecto a nivel de clúster.</p> <p>El intervalo después de una reducción fallida indica el período durante el cual el clúster no se puede escalar después de que el complemento de autoscaler active la reducción. Este intervalo tiene efecto a nivel de clúster.</p> <p><b>NOTA</b>                      Se recomienda no almacenar datos importantes en los nodos de un grupo de nodos porque después del ajuste automático, los datos no se pueden restaurar ya que los nodos se pueden eliminar.</p>

### Ajustes de cómputo

Puede configurar las especificaciones y el sistema operativo de un servidor en la nube, en el que se ejecutan sus aplicaciones en contenedores.

**Tabla 4-3** Parámetros de configuración

Parámetro	Descripción
Billing Mode	<p>Se admiten los siguientes modos de facturación:</p> <ul style="list-style-type: none"> <li>● Anual/Mensual                              Debe especificar la duración requerida si se selecciona <b>Yearly/Monthly</b>. Puede elegir si desea seleccionar <b>Auto-renew</b> según los requisitos del sitio. Su pedido se renovará automáticamente mensualmente o anualmente, dependiendo de si compró 1-9 meses, o 1-3 años.</li> <li>● Pago por uso                              Los recursos se facturarán en función de la duración del uso. Puede aprovisionar o eliminar recursos en cualquier momento.</li> </ul>

Parámetro	Descripción
Node Type	<p>Clúster de CCE:</p> <ul style="list-style-type: none"> <li>● ECS (VM): Los contenedores se ejecutan en ECS.</li> <li>● ECS (físico): Los contenedores se ejecutan en servidores que utilizan la arquitectura QingTian.</li> <li>● BMS: Los contenedores se ejecutan en BMS. Es necesario adjuntar los discos locales o los discos de EVS.</li> </ul> <p>Clúster de CCE Turbo:</p> <ul style="list-style-type: none"> <li>● ECS (VM): Los contenedores se ejecutan en ECS. Solo ECS de Trunkport (modelos que se pueden unir con múltiples interfaces de red elástica (ENI)) son compatibles.</li> <li>● ECS (físico): Los contenedores se ejecutan en servidores que utilizan la arquitectura QingTian.</li> </ul>
Container Engine	<p>Los clústeres de CCE admiten Docker y containerd en algunos escenarios.</p> <ul style="list-style-type: none"> <li>● Los nodos que ejecutan CentOS, Ubuntu y EulerOS 2.9 soportan containerd. Los nodos de Arm que ejecutan EulerOS 2.5 y EulerOS 2.8 no admiten containerd.</li> <li>● Los clústeres de red de VPC de v1.23 y versiones posteriores admiten containerd. Los clústeres de red de túneles de contenedores de v1.23.2-r0 y versiones posteriores admiten containerd.</li> <li>● Para un clúster de CCE Turbo, <b>Docker</b> y <b>containerd</b> son compatibles. Para obtener más información, véase <a href="#">Asignación entre los sistemas operativos de nodos y los motores de contenedores</a>.</li> </ul>
Specifications	<p>Seleccione una especificación de nodo basada en los requisitos de servicio. Las especificaciones de nodo disponibles varían dependiendo de las regiones o AZ. Para obtener más información, consulte la consola de CCE.</p>
OS	<p>Seleccione un tipo de SO. Diferentes tipos de nodos soportan los sistemas operativos diferentes. Para obtener más información, véase <a href="#">Especificaciones de nodos compatibles</a>.</p> <p><b>Public image:</b> Seleccione un sistema operativo para el nodo.</p> <p><b>Private image:</b> Puede utilizar las imágenes privadas. Para obtener más información sobre cómo crear una imagen privada, consulte <a href="#">Creación de una imagen de nodo de CCE personalizada</a>.</p>

Parámetro	Descripción
Login Mode	<ul style="list-style-type: none"> <li>● <b>Contraseña</b>                      El nombre de usuario predeterminado es <b>root</b>. Introduzca la contraseña para iniciar sesión en el nodo y confirme la contraseña. Asegúrese de recordar la contraseña, ya que la necesitará cuando inicie sesión en el nodo.</li> <li>● <b>Par de claves</b>                      Seleccione el par de claves utilizado para iniciar sesión en el nodo. Puede seleccionar una clave compartida.                       Se utiliza un par de claves para la autenticación de identidad cuando se inicia sesión de forma remota en un nodo. Si no hay ningún par de claves disponible, haga clic en <b>Create Key Pair</b>. Para obtener más información sobre cómo crear un par de claves, consulte <a href="#">Creación de un par de claves</a>.</li> </ul>

### Ajustes de almacenamiento

Configure los recursos de almacenamiento en un nodo para los contenedores que se ejecuta en él. Establezca el tamaño del disco según los requisitos del sitio.

**Tabla 4-4** Parámetros de configuración

Parámetro	Descripción
System Disk	<p>Disco del sistema utilizado por el sistema operativo del nodo. El valor oscila entre 40 GB y 1,024 GB. El valor predeterminado es 50 GB.</p> <p><b>Encryption:</b> La encriptación de disco de datos proporciona una protección potente para sus datos. Las instantáneas generadas a partir de discos cifrados y los discos creados con estas instantáneas heredan automáticamente la función de encriptación. <b>Esta función sólo está disponible en algunas regiones.</b></p> <ul style="list-style-type: none"> <li>● <b>Encryption</b> no está seleccionado de forma predeterminada.</li> <li>● Después de seleccionar <b>Encryption</b>, puede seleccionar una clave existente en el cuadro de diálogo que se muestra. Si no hay ninguna clave disponible, haga clic en <b>View Key List</b> para crear una clave. Una vez creada la clave, haga clic en el icono de actualización.</li> </ul>

Parámetro	Descripción
Data Disk	<p><b>Se requiere al menos un disco de datos</b> para el tiempo de ejecución de contenedor y kubelet. <b>El disco de datos no se puede eliminar ni desinstalar. De lo contrario, el nodo no estará disponible.</b></p> <ul style="list-style-type: none"> <li>● Primer disco de datos: utilizado para el tiempo de ejecución de contenedor y kubelet. El valor oscila entre 20 GB y 32,768 GB. El valor predeterminado es 100 GB</li> <li>● Para otros discos de datos, el valor oscila entre 10 GB y 32,768 GB. El valor predeterminado es 100 GB.</li> </ul> <p><b>Configuración avanzada</b></p> <p>Haga clic en <b>Expand</b> para establecer los siguientes parámetros:</p> <ul style="list-style-type: none"> <li>● <b>Allocate Disk Space:</b> Seleccione esta opción para definir el espacio de disco ocupado por el tiempo de ejecución de contenedor para almacenar los directorios de trabajo, los datos de imagen de contenedor y los metadatos de imagen. Para obtener más información acerca de cómo asignar espacio en disco de datos, consulte <a href="#">Asignación de espacio en disco de datos</a>.</li> <li>● <b>Encryption:</b> La encriptación de disco de datos proporciona una protección potente para sus datos. Las instantáneas generadas a partir de discos cifrados y los discos creados con estas instantáneas heredan automáticamente la función de encriptación. <b>Esta función sólo está disponible en algunas regiones.</b> <ul style="list-style-type: none"> <li>– <b>Encryption</b> no está seleccionado de forma predeterminada.</li> <li>– Después de seleccionar <b>Encryption</b>, puede seleccionar una clave existente en el cuadro de diálogo que se muestra. Si no hay ninguna clave disponible, haga clic en <b>View Key List</b> para crear una clave. Una vez creada la clave, haga clic en el icono de actualización.</li> </ul> </li> </ul> <p><b>Adición de varios discos de datos</b></p> <p>Se puede agregar un máximo de cuatro discos de datos. De forma predeterminada, los discos sin procesar se crean sin ningún procesamiento. También puede hacer clic en <b>Expand</b> y seleccionar cualquiera de las siguientes opciones:</p> <ul style="list-style-type: none"> <li>● <b>Default:</b> De forma predeterminada, se crea un disco sin procesar sin ningún procesamiento.</li> <li>● <b>Mount Disk:</b> El disco de datos está conectado a un directorio especificado.</li> <li>● <b>Use as PV:</b> aplicable a escenarios en los que hay un requisito de alto rendimiento en PVs. La etiqueta <b>node.kubernetes.io/local-storage-persistent</b> se agrega al nodo con el PV configurado. El valor es <b>linear</b> o <b>striped</b>.</li> <li>● <b>Use as ephemeral volume:</b> aplicable a escenarios en los que EmptyDir exige un alto rendimiento.</li> </ul>

Parámetro	Descripción
	<p><b>NOTA</b></p> <ul style="list-style-type: none"> <li>● Los PV locales solo se admiten cuando la versión del clúster es v1.21.2-r0 o posterior y la versión adicional más reciente es 2.1.23 o posterior. Se recomienda 2.1.23 o posterior.</li> <li>● Los EV locales solo se admiten cuando la versión del clúster es v1.21.2-r0 o posterior y la versión adicional es 1.2.29 o posterior.</li> </ul> <p><b>PV local</b> y <b>EV local</b> soportan los siguientes modos de escritura:</p> <ul style="list-style-type: none"> <li>● <b>Linear</b>: Un volumen lógico lineal integra uno o más volúmenes físicos. Los datos se escriben en el siguiente volumen físico cuando se agota el anterior.</li> <li>● <b>Striped</b>: Un volumen lógico rayado separa los datos en bloques del mismo tamaño y los almacena en múltiples volúmenes físicos en secuencia, lo que permite que los datos se lean y escriban simultáneamente. No se puede ampliar un grupo de almacenamiento compuesto por volúmenes seccionados. Esta opción solo se puede seleccionar cuando existen varios volúmenes.</li> </ul> <p><b>Descripción de disco local</b></p> <p>Si la variante de nodo es con uso intensivo de disco o con capacidad ultraalta de E/S, un disco de datos puede ser un disco local.</p> <p>Los discos locales pueden descomponerse y no garantizar la fiabilidad de los datos. Se recomienda almacenar los datos de servicio en los discos de EVS, que son más fiables que los discos locales.</p>

### Ajustes de redes

Configure los recursos de red para permitir el acceso a nodos y aplicaciones en contenedores.

**Tabla 4-5** Parámetros de configuración

Parámetro	Descripción
Node Subnet	La subred de nodo seleccionada durante la creación del clúster se utiliza de forma predeterminada. Puede elegir otra subred en su lugar.
Node IP	Se admite la asignación aleatoria.
Associate Security Group	<p>Grupo de seguridad utilizado por los nodos creados en el grupo de nodos. Se puede seleccionar un máximo de 5 grupos de seguridad.</p> <p>Cuando se crea un clúster, se crea un grupo de seguridad de nodo denominado <b>{Cluster name}-cce-node-{Random ID}</b> y se utiliza de forma predeterminada.</p> <p>El tráfico necesita pasar con ciertos puertos en el grupo de seguridad de nodo para garantizar las comunicaciones de nodo. Asegúrese de haber habilitado estos puertos si selecciona otro grupo de seguridad. Para obtener más información, vea <a href="#">Configuración de reglas de grupo de seguridad de clúster</a>.</p>

### Configuración avanzada

Configure las capacidades avanzadas de nodo como etiquetas, manchas y comandos de inicio.

**Tabla 4-6** Parámetros de configuración avanzadas

Parámetro	Descripción
Kubernetes Label	<p>Haga clic en <b>Add Label</b> para establecer el par clave-valor asociado a los objetos de Kubernetes (como los pods). Se puede agregar un máximo de 20 etiquetas.</p> <p>Las etiquetas se pueden utilizar para distinguir nodos. Con la configuración de afinidad de carga de trabajo, los pods de contenedor se pueden programar en un nodo específico. Para obtener más información, consulte <a href="#">Etiquetas y selectores</a>.</p>
Resource Tag	<p>Puede agregar etiquetas de recursos para clasificar recursos.</p> <p>Puede crear <b>predefined tags</b> en Tag Management Service (TMS). Las etiquetas predefinidas son visibles para todos los recursos de servicio que admiten la función de etiquetado. Puede utilizar estas etiquetas para mejorar la eficiencia del etiquetado y la migración de recursos. Para obtener más información, consulte <a href="#">Creación de etiquetas predefinidas</a>.</p> <p>CCE creará automáticamente la etiqueta "CCE-Dynamic-Provisioning-Node=<i>node id</i>".</p>
Taint	<p>Este parámetro se deja en blanco por defecto. Puede agregar manchas para establecer antiafinidad para el nodo. Se permite un máximo de 10 manchas para cada nodo. Cada mancha contiene los siguientes parámetros:</p> <ul style="list-style-type: none"> <li>● <b>Key</b>: Una clave debe contener de 1 a 63 caracteres, comenzando por una letra o un dígito. Solo se permiten letras, dígitos, guiones (-), guiones bajos (_) y puntos (.). Un nombre de subdominio de DNS se puede utilizar como prefijo de una clave.</li> <li>● <b>Value</b>: Un valor debe comenzar con una letra o un dígito y puede contener un máximo de 63 caracteres, incluidos letras, dígitos, guiones (-), guiones bajos (_) y puntos (.).</li> <li>● <b>Effect</b>: Las opciones disponibles son <b>NoSchedule</b>, <b>PreferNoSchedule</b> y <b>NoExecute</b>.</li> </ul> <p>Para obtener más información, véase <a href="#">Gestión de manchas de nodos</a>.</p> <p><b>NOTA</b>                      Para un clúster de v1.19 o anterior, es posible que la carga de trabajo se haya programado en un nodo antes de agregar la mancha. Para evitar tal situación, seleccione un clúster de v1.19 o posterior.</p>
Max. Pods	<p>Número máximo de pods que se pueden ejecutar en el nodo, incluidos los pods del sistema predeterminados. Rango de valores: 16 a 256</p> <p>Este límite evita que el nodo se sobrecargue con pods.</p> <p>Este número también se decide por otros factores. Para obtener más información, véase <a href="#">Número máximo de pods que se pueden crear en un nodo</a>.</p>

Parámetro	Descripción
ECS Group	<p>Un grupo de ECS agrupa lógicamente ECS. Los ECS del mismo grupo de ECS cumplen con la misma política asociada con el grupo de ECS.</p> <p><b>Anti-affinity:</b> los ECS de un grupo de ECS se despliegan en diferentes hosts físicos para mejorar la confiabilidad del servicio.</p> <p>Seleccione un grupo de ECS existente o haga clic en <b>Add ECS Group</b> para crear uno. Una vez creado el grupo de ECS, haga clic en el botón de actualizar.</p>
Pre-installation Command	<p>Ingrese los comandos. Se permite un máximo de 1,000 caracteres.</p> <p>El script se ejecutará antes de que se instale el software de Kubernetes. Tenga en cuenta que si el script es incorrecto, es posible que el software de Kubernetes no se instale.</p>
Post-installation Command	<p>Ingrese los comandos. Se permite un máximo de 1,000 caracteres.</p> <p>El script se ejecutará después de instalar el software de Kubernetes y no afectará a la instalación.</p>
Agency	<p>El administrador de la cuenta crea una delegación en la consola de IAM. Al crear una delegación, puede compartir sus recursos de servidor en la nube con otra cuenta, o confiar a una persona o equipo más profesional para gestionar sus recursos.</p> <p>Si no hay ninguna delegación disponible, haga clic en <b>Create Agency</b> a la derecha para crear una.</p>

**Paso 4** Haga clic en **Next: Confirm**. Asegúrese de haber leído y entendido la [Declaración del Image Management Service](#).

**Paso 5** Haga clic en **Submit**.

---Fin

## 4.3 Gestión de un grupo de nodos

### 4.3.1 Configuración de un grupo de nodos

#### Notas y restricciones

El grupo de nodos predeterminado de DefaultPool no admite las siguientes operaciones de gestión.

#### Gestión de configuraciones

CCE le permite personalizar altamente la configuración de parámetros de Kubernetes en los componentes principales de un clúster. Para obtener más información, consulte [kubelet](#).

Esta función solo se admite en los clústeres de **v1.15 y posterior**. No se muestra para clústeres anteriores a v1.15.

- Paso 1** Inicie sesión en la consola de CCE.
- Paso 2** Haga clic en el nombre del clúster y acceda a la consola del clúster. Elija **Nodos** en el panel de navegación y haga clic en la ficha **Node Pools** de la derecha.
- Paso 3** Elija **More > Manage** junto al nombre del grupo de nodos.
- Paso 4** En la página **Manage Component** de la derecha, cambie los valores de los siguientes parámetros de Kubernetes:

**Tabla 4-7** kubelet

Parámetro	Descripción	Valor predeterminado	Notas
cpu-manager-policy	<p>Especifica la configuración de enlace del núcleo de CPU. Para obtener más información, véase <a href="#">Programación de CPU</a>.</p> <ul style="list-style-type: none"> <li>● <b>none</b>: desactiva los pods para que no ocupen exclusivamente las CPU. Seleccione este valor si desea un grupo grande de núcleos de CPU compatibles.</li> <li>● <b>static</b>: permite que los pods ocupen exclusivamente las CPU. Seleccione este valor si su carga de trabajo es sensible a la latencia en la memoria caché y la programación de la CPU.</li> <li>● <b>enhanced-static</b>: permite que los pods explosibles utilicen preferentemente núcleos de CPU. Seleccione este valor si su carga de trabajo tiene una gran diferencia de picos y se encuentra en el estado valle la mayor parte del tiempo.</li> </ul>	Ninguno	Los valores se pueden modificar durante el ciclo de vida del grupo de nodos.

Parámetro	Descripción	Valor predeterminado	Notas
kube-api-qps	Consulta por segundo (QPS) para usar mientras se habla con kube-apiserver.	100	
kube-api-burst	Ráfaga para usar mientras se habla con kube-apiserver.	100	
max-pods	Número máximo de pods gestionados por kubelet.	40 20	
pod-pids-limit	Límite de PID en Kubernetes	-1	
with-local-dns	Si se utiliza la dirección IP local como ClusterDNS del nodo.	false	
event-qps	Límite de QPS para la creación de eventos	5	
allowed-unsafe-sysctls	<p>Se permite la configuración del sistema insegura.</p> <p>A partir de <b>v1.17.17</b>, CCE habilita políticas de seguridad de pod para kube-apiserver. Es necesario agregar las configuraciones correspondientes a <b>allowedUnsafeSysctls</b> de una política de seguridad de pod para que la política entre en vigor. (Esta configuración no es necesaria para clústeres anteriores a v1.17.17.) Para obtener más información, véase <a href="#">Ejemplo de habilitación de Sysctls inseguros en la Política de Seguridad de Pods</a>.</p>	[]	

Parámetro	Descripción	Valor predeterminado	Notas
over-subscription-resource	<p>Si se debe habilitar la sobresuscripción de nodo.</p> <p>Si este parámetro se establece en <b>true</b>, se activa la función de sobresuscripción de nodo. Para obtener más información, véase <a href="#">Despliegue híbrido de trabajos en línea y fuera de línea</a>.</p>	true	-
colocation	<p>Si habilitar los despliegue híbridos de nodo.</p> <p>Si este parámetro se establece en <b>true</b> se activa la función del despliegue híbrido de nodo. Para obtener más información, véase <a href="#">Despliegue híbrido de trabajos en línea y fuera de línea</a>.</p>	true	-
kube-reserved-mem system-reserved-mem	Memoria de nodo reservada.	Depende de las especificaciones del nodo. Para obtener más información, véase <a href="#">Descripción de los recursos de nodos reservados</a> .	La suma de kube-reserved-mem y system-reserved-mem es menos de la mitad de la memoria.

Parámetro	Descripción	Valor predeterminado	Notas
topology-manager-policy	<p>Establezca la política de gestión de topología.</p> <p>Los valores válidos son los siguientes:</p> <ul style="list-style-type: none"> <li>● <b>restricted</b>: kubelet solo acepta pods que logran una alineación de NUMA óptima en los recursos solicitados.</li> <li>● <b>best-effort</b>: kubelet selecciona preferentemente pods que implementan la alineación de NUMA en recursos de CPU y dispositivos.</li> <li>● <b>none</b> (predeterminado): La política de gestión de topología está deshabilitada.</li> <li>● <b>single-numa-node</b>: kubelet solo permite pods que están alineados con el mismo nodo de NUMA en los términos de recursos de CPU y dispositivo.</li> </ul>	Ninguno	<p>Los valores se pueden modificar durante el ciclo de vida del grupo de nodos.</p> <p><b>AVISO</b></p> <p>Tenga cuidado al modificar topology-manager-policy y topology-manager-scope reiniciará kubelet y volverá a calcular la asignación de recursos de los pods según la política modificada. Como resultado, los pods en ejecución pueden reiniciarse o incluso no recibir recursos.</p>
topology-manager-scope	<p>Establezca la granularidad de alineación de recursos de la política de gestión de topología. Los valores válidos son los siguientes:</p> <ul style="list-style-type: none"> <li>● <b>container</b> (predeterminado)</li> <li>● <b>pod</b></li> </ul>	container	
resolv-conf	<p>Archivo de configuración de resolución DNS especificado por el contenedor</p>	El valor predeterminado es null.	-

**Tabla 4-8** kube-proxy

Parámetro	Descripción	Valor predeterminado	Notas
conntrack-min	sysctl -w net.nf_conntrack_max	131072	Los valores se pueden modificar durante el ciclo de vida del grupo de nodos.
conntrack-tcp-timeout-close-wait	sysctl -w net.netfilter.nf_conntrack_tcp_timeout_close_wait	1h0m0s	

**Tabla 4-9** Componentes de red (disponibles solo para los clústeres de CCE Turbo)

Parámetro	Descripción	Valor predeterminado	Notas
nic-threshold	Umbral bajo del número de ENIs unidos: Umbral alto del número de ENIs unidos <b>NOTA</b> Este parámetro está siendo descartado. Utilice los parámetros dinámicos de preenlace de los otros cuatro ENI.	Default: 0:0	-
nic-minimum-target	Número mínimo de ENIs enlazados a los nodos del grupo de nodos	Default: 10	-
nic-maximum-target	Número máximo de ENI preenlazados a un nodo en el nivel de grupo de nodos	Default: 0	-
nic-warm-target	Número de ENI preenlazados a un nodo en el nivel de grupo de nodos	Default: 2	-
nic-max-above-warm-target	Recupere el número de ENI preenlazados a un nodo en el nivel de grupo de nodos	Default: 2	-

**Tabla 4-10** Grupo de seguridad de pod en un grupo de nodos (disponible solo para clústeres de CCE Turbo)

Parámetro	Descripción	Valor predeterminado	Notas
security_groups_for_nodepool	<ul style="list-style-type: none"> <li>● Grupo de seguridad predeterminado utilizado por los pods en un grupo de nodos. Puede introducir el ID del grupo de seguridad. Si este parámetro no está definido, se utiliza el grupo de seguridad predeterminado de la red de contenedor del clúster. Se puede especificar un máximo de cinco ID de grupo de seguridad al mismo tiempo, separados por punto y coma (;).</li> <li>● La prioridad del grupo de seguridad es menor que la del grupo de seguridad configurado para <a href="#">Security Groups</a>.</li> </ul>	-	-

**Tabla 4-11** Docker (disponible solo para los grupos de nodos que usan Docker)

Parámetro	Descripción	Valor predeterminado	Notas
native-umask	`--exec-opt native.umask`	normal	No se puede cambiar.
docker-base-size	`--storage-opt dm.basesize`	0	No se puede cambiar.
insecure-registry	Dirección de un registro de imágenes inseguro	false	No se puede cambiar.

Parámetro	Descripción	Valor predeterminado	Notas
limitcore	Tamaño máximo de un archivo de núcleo en un contenedor. La unidad es byte. Si no se especifica, el valor es <b>infinity</b> .	5368709120	-
default-ulimit-nofile	Límite en el número de manejadores en un contenedor	{soft}:{hard}	El valor no puede exceder el valor del parámetro del núcleo <b>nr_open</b> y no puede ser un número negativo. Puede ejecutar el siguiente comando para obtener el parámetro del kernel <b>nr_open</b> : <pre>sysctl -a   grep nr_open</pre>

**Tabla 4-12** containerd (disponible solo para los grupos de nodos que usan containerd)

Parámetro	Descripción	Valor predeterminado	Notas
devmapper-base-size	Espacio de datos disponible de un solo contenedor	-	No se puede cambiar.
limitcore	Tamaño máximo de un archivo de núcleo en un contenedor. La unidad es byte. Si no se especifica, el valor es <b>infinity</b> .	5368709120	-
default-ulimit-nofile	Límite en el número de manejadores en un contenedor	1048576	El valor no puede exceder el valor del parámetro del núcleo <b>nr_open</b> y no puede ser un número negativo. Puede ejecutar el siguiente comando para obtener el parámetro del kernel <b>nr_open</b> : <pre>sysctl -a   grep nr_open</pre>

**Paso 5** Haga clic en **OK**.

---Fin

## 4.3.2 Actualización de un grupo de nodos

### Restricciones

- Solo los clústeres de v1.19 o posterior admiten la modificación del motor contenedor, el SO, el tamaño del sistema/disco de datos, la asignación de espacio en disco de datos y la configuración de scripts de preinstalación/postinstalación.
- Al editar el motor contenedor, el sistema operativo, los scripts previos y posteriores a la instalación y las etiquetas de recursos del grupo de nodos. La configuración modificada solo tiene efecto para nodos nuevos. Para sincronizar la configuración con los nodos existentes, debe restablecer manualmente los nodos existentes.
- La modificación de la asignación de espacio en disco de datos y el tamaño de disco de sistema/datos de un grupo de nodos tiene efecto solo para los nodos nuevos. La configuración no se puede sincronizar aunque se restablezcan los nodos existentes.
- Las actualizaciones de las etiquetas y manchas de kubernetes se sincronizan automáticamente con los nodos existentes. No es necesario restablecer los nodos.

### Edición de un grupo de nodos

**Paso 1** Inicie sesión en la consola de CCE.

**Paso 2** Haga clic en el nombre del clúster y acceda a la consola del clúster. Elija **Nodos** en el panel de navegación y haga clic en la ficha **Node Pools** de la derecha.

**Paso 3** Haga clic en **Edit** junto al nombre del grupo de nodos que va a editar. Edite los parámetros en la página **Edit Node Pool** mostrada.

#### Ajustes básicos

**Tabla 4-13** Configuración básica

Parámetro	Descripción
Node Pool Name	Nombre del grupo de nodos.

Parámetro	Descripción
Auto Scaling	<p>De forma predeterminada, este parámetro está deshabilitado. Después de habilitar el escalador automático haciendo clic en , los nodos del grupo de nodos se crean o eliminan automáticamente en función de los requisitos de servicio.</p> <ul style="list-style-type: none"> <li>● <b>Maximum Nodes</b> y <b>Minimum Nodes</b>: Puede establecer el número máximo y mínimo de nodos para asegurarse de que el número de nodos a escalar está dentro de un rango adecuado.</li> <li>● <b>Priority</b>: Un valor mayor indica una prioridad más alta. Por ejemplo, si este parámetro se establece en <b>1</b> y <b>4</b> respectivamente para los grupos de nodos A y B, B tiene una prioridad más alta que A, y el ajuste automático se activa primero para B. Si las prioridades de múltiples grupos de nodos se establecen en el mismo valor, por ejemplo, el <b>2</b>, los grupos de nodos no se priorizan y el sistema realiza el ajuste basándose en el principio mínimo de desperdicio de recursos. Después de actualizar la prioridad, la configuración entra en vigor en 1 minuto.</li> <li>● <b>Cooldown Period</b>: Ingrese un período, en minutos. Este campo indica el período durante el cual los nodos agregados en el grupo de nodos actual no se pueden escalar.</li> </ul> <p>Si el campo <b>Autoscaler</b> está establecido en On, instale el <a href="#">autoscaler</a> para utilizar la función autoscaler.</p>

### Ajustes de cómputo

Tabla 4-14 Parámetros de configuración

Parámetro	Descripción
AZ	<p>La zona de disponibilidad donde se encuentra el nodo. Los nodos de un clúster se pueden crear en las diferentes AZ para una mayor fiabilidad. El valor no se puede cambiar después de la creación.</p> <p>Se recomienda seleccionar <b>Random</b> para desplegar su nodo en una AZ aleatoria basada en la especificación de nodo seleccionada.</p> <p>Una AZ es una región física donde los recursos utilizan las fuentes de alimentación y las redes independientes. Las AZ están físicamente aisladas, pero se interconectan a través de una red interna. Para mejorar la disponibilidad de la carga de trabajo, cree nodos en las diferentes AZ.</p> <p><b>NOTA</b>                      La modificación de la configuración de AZ solo tiene efecto para los nuevos nodos. No se pueden modificar las AZ de los nodos existentes.</p>

Parámetro	Descripción
Node Type	<p>Clúster de CCE:</p> <ul style="list-style-type: none"> <li>● ECS (VM): Los contenedores se ejecutan en ECS.</li> <li>● ECS (físico): Los contenedores se ejecutan en servidores que utilizan la arquitectura QingTian.</li> <li>● BMS: Los contenedores se ejecutan en BMS. Es necesario adjuntar los discos locales o los discos de EVS.</li> </ul> <p>Clúster de CCE Turbo:</p> <ul style="list-style-type: none"> <li>● ECS (VM): Los contenedores se ejecutan en ECS. Solo ECS de Trunkport (modelos que se pueden unir con múltiples interfaces de red elástica (ENI)) son compatibles.</li> <li>● ECS (físico): Los contenedores se ejecutan en servidores que utilizan la arquitectura QingTian.</li> </ul> <p><b>NOTA</b>                      Esta configuración no se puede modificar ahora.</p>
Container Engine	<p>Los clústeres de CCE admiten Docker y containerd en algunos escenarios.</p> <ul style="list-style-type: none"> <li>● Los nodos que ejecutan CentOS, Ubuntu y EulerOS 2.9 soportan containerd. Los nodos de Arm que ejecutan EulerOS 2.5 y EulerOS 2.8 no admiten containerd.</li> <li>● Los clústeres de red de VPC de v1.23 y versiones posteriores admiten containerd. Los clústeres de red de túneles de contenedores de v1.23.2-r0 y versiones posteriores admiten containerd.</li> <li>● Para un clúster de CCE Turbo, <b>Docker</b> y <b>containerd</b> son compatibles. Para obtener más información, véase <a href="#">Asignación entre los sistemas operativos de nodos y los motores de contenedores</a>.</li> </ul> <p><b>NOTA</b>                      Después de modificar el motor contenedor, la modificación se aplica automáticamente cuando se agrega un nodo. Para los nodos existentes, debe restablecer manualmente los nodos para que la modificación surta efecto.</p>
Specifications	<p>Seleccione las especificaciones de nodo que mejor se adapten a las necesidades de su negocio.</p>
OS	<p>Seleccione un tipo de sistema operativo. Diferentes tipos de nodos soportan los sistemas operativos diferentes. Para obtener más información, véase <a href="#">Especificaciones de nodos compatibles</a>.</p> <p><b>Public image:</b> Seleccione un sistema operativo para el nodo.</p> <p><b>Private image:</b> Puede utilizar las imágenes privadas. Para obtener más información sobre cómo crear una imagen privada, consulte <a href="#">Creación de una imagen de nodo de CCE personalizada</a>.</p> <p><b>NOTA</b>                      Una vez modificado el sistema operativo, la modificación tiene efecto automáticamente cuando se agrega un nodo. Para los nodos existentes, debe restablecer manualmente los nodos para que la modificación surta efecto.</p>

### Ajustes de almacenamiento

 **NOTA**

La modificación de la configuración de almacenamiento solo tiene efecto para los nuevos nodos. La configuración no se puede sincronizar aunque se restablezcan los nodos existentes.

**Tabla 4-15** Parámetros de configuración

Parámetro	Descripción
System Disk	<p>Disco del sistema utilizado por el sistema operativo del nodo. El valor oscila entre 40 GB y 1,024 GB. El valor predeterminado es 50 GB.</p> <p><b>Encryption:</b> La encriptación de disco de datos proporciona una protección potente para sus datos. Las instantáneas generadas a partir de discos cifrados y los discos creados con estas instantáneas heredan automáticamente la función de encriptación. <b>Esta función sólo está disponible en algunas regiones.</b></p> <ul style="list-style-type: none"><li>● <b>Encryption</b> no está seleccionado de forma predeterminada.</li><li>● Después de seleccionar <b>Encryption</b>, puede seleccionar una clave existente en el cuadro de diálogo que se muestra. Si no hay ninguna clave disponible, haga clic en <b>View Key List</b> para crear una clave. Una vez creada la clave, haga clic en el icono de actualización.</li></ul>

Parámetro	Descripción
Data Disk	<p><b>Se requiere al menos un disco de datos</b> para el tiempo de ejecución de contenedor y kubelet. <b>El disco de datos no se puede eliminar ni desinstalar. De lo contrario, el nodo no estará disponible.</b></p> <ul style="list-style-type: none"> <li>● Primer disco de datos: utilizado para el tiempo de ejecución de contenedor y kubelet. El valor oscila entre 20 GB y 32,768 GB. El valor predeterminado es 100 GB</li> <li>● Para otros discos de datos, el valor oscila entre 10 GB y 32,768 GB. El valor predeterminado es 100 GB.</li> </ul> <p><b>Configuración avanzada</b></p> <p>Haga clic en <b>Expand</b> para establecer los siguientes parámetros:</p> <ul style="list-style-type: none"> <li>● <b>Allocate Disk Space:</b> Seleccione esta opción para definir el espacio de disco ocupado por el tiempo de ejecución de contenedor para almacenar los directorios de trabajo, los datos de imagen de contenedor y los metadatos de imagen. Para obtener más información acerca de cómo asignar espacio en disco de datos, consulte <a href="#">Asignación de espacio en disco de datos</a>.</li> <li>● <b>Encryption:</b> La encriptación de disco de datos proporciona una protección potente para sus datos. Las instantáneas generadas a partir de discos cifrados y los discos creados con estas instantáneas heredan automáticamente la función de encriptación. <b>Esta función sólo está disponible en algunas regiones.</b> <ul style="list-style-type: none"> <li>– <b>Encryption</b> no está seleccionado de forma predeterminada.</li> <li>– Después de seleccionar <b>Encryption</b>, puede seleccionar una clave existente en el cuadro de diálogo que se muestra. Si no hay ninguna clave disponible, haga clic en <b>View Key List</b> para crear una clave. Una vez creada la clave, haga clic en el icono de actualización.</li> </ul> </li> </ul> <p><b>Adición de varios discos de datos</b></p> <p>Se puede agregar un máximo de cuatro discos de datos. De forma predeterminada, los discos sin procesar se crean sin ningún procesamiento. También puede hacer clic en <b>Expand</b> y seleccionar cualquiera de las siguientes opciones:</p> <ul style="list-style-type: none"> <li>● <b>Default:</b> De forma predeterminada, se crea un disco sin procesar sin ningún procesamiento.</li> <li>● <b>Mount Disk:</b> El disco de datos está conectado a un directorio especificado.</li> <li>● <b>Use as PV:</b> aplicable a escenarios en los que hay un requisito de alto rendimiento en PVs. La etiqueta <b>node.kubernetes.io/local-storage-persistent</b> se agrega al nodo con el PV configurado. El valor es <b>linear</b> o <b>striped</b>.</li> <li>● <b>Use as ephemeral volume:</b> aplicable a escenarios en los que EmptyDir exige un alto rendimiento.</li> </ul>

Parámetro	Descripción
	<p><b>NOTA</b></p> <ul style="list-style-type: none"> <li>● Los PV locales solo se admiten cuando la versión del clúster es v1.21.2-r0 o posterior y la versión adicional más reciente es 2.1.23 o posterior. Se recomienda 2.1.23 o posterior.</li> <li>● Los EV locales solo se admiten cuando la versión del clúster es v1.21.2-r0 o posterior y la versión adicional es 1.2.29 o posterior.</li> </ul> <p><b>PV local</b> y <b>EV local</b> soportan los siguientes modos de escritura:</p> <ul style="list-style-type: none"> <li>● <b>Linear</b>: Un volumen lógico lineal integra uno o más volúmenes físicos. Los datos se escriben en el siguiente volumen físico cuando se agota el anterior.</li> <li>● <b>Striped</b>: Un volumen lógico rayado separa los datos en bloques del mismo tamaño y los almacena en múltiples volúmenes físicos en secuencia, lo que permite que los datos se lean y escriban simultáneamente. No se puede ampliar un grupo de almacenamiento compuesto por volúmenes seccionados. Esta opción solo se puede seleccionar cuando existen varios volúmenes.</li> </ul> <p><b>Descripción de disco local</b></p> <p>Si la variante de nodo es con uso intensivo de disco o con capacidad ultraalta de E/S, un disco de datos puede ser un disco local.</p> <p>Los discos locales pueden descomponerse y no garantizar la fiabilidad de los datos. Se recomienda almacenar los datos de servicio en los discos de EVS, que son más fiables que los discos locales.</p>

### Configuración avanzada

Tabla 4-16 Ajustes avanzados

Parámetro	Descripción
Kubernetes Label	<p>Haga clic en <b>Add Label</b> para establecer el par clave-valor asociado a los objetos de Kubernetes (como los pods). Se puede agregar un máximo de 20 etiquetas.</p> <p>Las etiquetas se pueden utilizar para distinguir nodos. Con la configuración de afinidad de carga de trabajo, los pods de contenedor se pueden programar en un nodo específico. Para obtener más información, consulte <a href="#">Etiquetas y selectores</a>.</p> <p><b>NOTA</b></p> <p>Después de que se modifica una <b>Kubernetes label</b>, los nodos de inventario en el grupo de nodos se actualizan sincrónicamente.</p>

Parámetro	Descripción
Resource Tag	<p>Puede agregar etiquetas de recursos para clasificar recursos.</p> <p>Puede crear <b>predefined tags</b> en Tag Management Service (TMS). Las etiquetas predefinidas son visibles para todos los recursos de servicio que admiten la función de etiquetado. Puede utilizar estas etiquetas para mejorar la eficiencia del etiquetado y la migración de recursos. Para obtener más información, consulte <a href="#">Creación de etiquetas predefinidas</a>.</p> <p>CCE creará automáticamente la etiqueta "CCE-Dynamic-Provisioning-Node=<i>node id</i>".</p> <p><b>NOTA</b>                      Después de modificar una <b>resource tag</b>, la modificación tiene efecto automáticamente cuando se agrega un nodo. Para los nodos existentes, debe restablecer manualmente los nodos para que la modificación surta efecto.</p>
Taint	<p>Este campo se deja en blanco por defecto. Puede agregar manchas para establecer antiafinidad para el nodo. Se permite un máximo de 10 manchas para cada nodo. Cada mancha contiene los siguientes parámetros:</p> <ul style="list-style-type: none"> <li>● <b>Key:</b> Una clave debe contener de 1 a 63 caracteres, comenzando por una letra o un dígito. Solo se permiten letras, dígitos, guiones (-), guiones bajos (_) y puntos (.). Un nombre de subdominio de DNS se puede utilizar como prefijo de una clave.</li> <li>● <b>Value:</b> Un valor debe comenzar con una letra o un dígito y puede contener un máximo de 63 caracteres, incluidos letras, dígitos, guiones (-), guiones bajos (_) y puntos (.).</li> <li>● <b>Effect:</b> Las opciones disponibles son <b>NoSchedule</b>, <b>PreferNoSchedule</b> y <b>NoExecute</b>.</li> </ul> <p>Para obtener más información, véase <a href="#">Gestión de manchas de nodos</a>.</p> <p><b>NOTA</b>                      Después de que se modifica una <b>taint</b>, los nodos de inventario en el grupo de nodos se actualizan sincrónicamente.</p>
Edit Key pair	<p>Solo los grupos de nodos que utilizan pares de claves para iniciar sesión admiten la edición de pares de claves. Puede seleccionar otro par de claves.</p> <p><b>NOTA</b>                      El par de claves editado tiene efecto automáticamente cuando se agrega un nodo. Para los nodos existentes, es necesario restablecer manualmente los nodos para que el par de claves surta efecto.</p>

Parámetro	Descripción
Pre-installation Command	<p>Ingrese los comandos. Se permite un máximo de 1,000 caracteres.</p> <p>El script se ejecutará antes de que se instale el software de Kubernetes. Tenga en cuenta que si el script es incorrecto, es posible que el software de Kubernetes no se instale.</p> <p><b>NOTA</b>                      El comando modificado de preinstalación tiene efecto automáticamente cuando se agrega un nodo. Para los nodos existentes, debe restablecer manualmente los nodos para que la modificación surta efecto.</p>
Post-installation Command	<p>Ingrese los comandos. Se permite un máximo de 1,000 caracteres.</p> <p>El script se ejecutará después de instalar el software de Kubernetes y no afectará a la instalación.</p> <p><b>NOTA</b>                      El comando modificado después de la instalación tiene efecto automáticamente cuando se agrega un nodo. Para los nodos existentes, debe restablecer manualmente los nodos para que la modificación surta efecto.</p>

**Paso 4** Cuando se complete la configuración, haga clic en **OK**.

Después de actualizar los parámetros del grupo de nodos, vaya a la página **Nodos** para comprobar si el nodo al que pertenece el grupo de nodos está actualizado. Puede restablecer el nodo para sincronizar las actualizaciones de configuración para el grupo de nodos.

### AVISO

La modificación de la configuración del sistema/disco de datos de un grupo de nodos solo tiene efecto para nuevos nodos. La configuración no se puede sincronizar aunque se restablezcan los nodos existentes.



---Fin

## 4.3.3 Escalamiento de un grupo de nodos

Puede especificar una especificación en un grupo de nodos para escalar.

### AVISO

El grupo de nodos predeterminado no admite el ajuste. Utilice [Creación de un nodo](#) para agregar un nodo.

**Paso 1** Inicie sesión en la consola de CCE.

**Paso 2** Haga clic en el nombre del clúster y acceda a la consola del clúster. Elija **Nodes** en el panel de navegación y haga clic en la ficha **Node Pools** de la derecha.

**Paso 3** Elija **More > Scaling** en la columna **Operation** del grupo de nodos de destino.

**Paso 4** En la ventana **Node Pool Scaling** mostrada, defina los parámetros de ajuste.

- **Number of Scaling Targets:** El número de nodos de destino no puede exceder la escala de gestión del clúster actual.
- **Node Configuration:** Utilice la especificación seleccionada para ampliar la capacidad de un nodo. Si la especificación se agota, la expansión de la capacidad fallará.

#### **NOTA**

- Si el número de nodos de destino es mayor que el número actual, se agregarán algunos nodos. Si el número de nodos de destino es menor que el número actual, se eliminarán algunos nodos.
- Durante la reducción de capacidad, si el número de nodos de la especificación especificada es menor que el número de nodos a eliminar, los nodos de otras especificaciones se eliminarán.

### Node Pool Scaling

Node Pool Name	 -nodepool-72676
Billing Mode	Pay-per-use
Current Quantity	1
Number of Scaling Targets	<input type="button" value="-"/> <input type="text" value="1"/> <input type="button" value="+"/> 
Node Configuration	<input type="text" value="c7.large.2   可用区3"/>

Use the selected flavor to expand the capacity of a node. If the flavor is sold out, the capacity expansion will fail.

**Paso 5** Haga clic en **OK**.

----**Fin**

## 4.3.4 Copia de un grupo de nodos

Puede copiar la configuración de un grupo de nodos existente para crear un nuevo grupo de nodos en la consola de CCE.

**Paso 1** Inicie sesión en la consola de CCE.

**Paso 2** Haga clic en el nombre del clúster y acceda a la consola del clúster. Elija **Nodes** en el panel de navegación y haga clic en la ficha **Node Pools** de la derecha.

**Paso 3** Elija **More > Copy** junto al nombre de un grupo de nodos para copiar el grupo de nodos.



**Paso 4** Las configuraciones del grupo de nodos seleccionado se replican en la página **Clone Node Pool**. Puede editar las configuraciones según sea necesario. Para obtener más información sobre los elementos de configuración, consulte [Creación de un grupo de nodos](#). Después de confirmar la configuración, haga clic en **Next: Confirm**.

**Paso 5** En la página **Confirm**, confirme la configuración del grupo de nodos y haga clic en **Submit**. A continuación, se crea un nuevo grupo de nodos basado en la configuración editada.

----Fin

### 4.3.5 Migración de un nodo

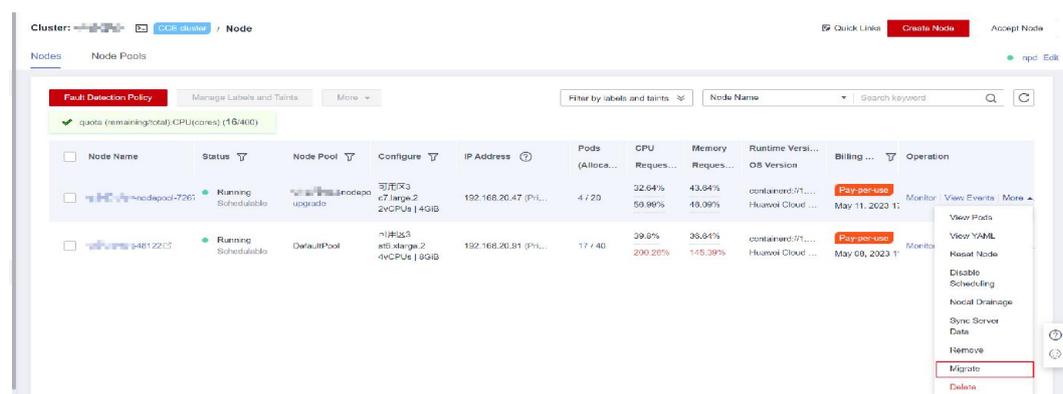
Los nodos de un grupo de nodos se pueden migrar. Actualmente, los nodos de un grupo de nodos solo se pueden migrar al grupo de nodos predeterminado (grupo predeterminado) en el mismo clúster.

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** En el panel de navegación, elija **Nodes** y cambie a la página de ficha **Node Pools**.

**Paso 3** Haga clic en **View Node** en la columna **Operation** del grupo de nodos que se va a migrar.

**Paso 4** Haga clic en **More > Migrate** en la columna **Operation** del nodo de destino para migrar el nodo.



**Paso 5** En la ventana **Migrate Node** mostrada, confirme la información.

#### 📖 NOTA

La migración no afecta a las etiquetas de recursos originales, las etiquetas de Kubernetes y las manchas del nodo.

----Fin

## 4.3.6 Eliminación de un pool de nodos

Al eliminar un grupo de nodos se eliminarán los nodos del grupo. Los pods de estos nodos se migrarán automáticamente a los nodos disponibles en otros grupos de nodos.

### Notas y restricciones

- No se puede eliminar un grupo de nodos de facturación anual/mensual antes de que se eliminen primero todos los nodos.
- La eliminación de un nodo causará la pérdida de datos de PVC/PV para los **PV locales** asociados con el nodo. Estos PVC y PV no se pueden restaurar o utilizar de nuevo. En este escenario, el pod que utiliza el PV local se desaloja del nodo. Se crea un nuevo pod y permanece en el estado pendiente. Esto se debe a que el PVC utilizado por el pod tiene una etiqueta de nodo, debido a lo cual el pod no se puede programar.

### Precauciones

- Al eliminar un grupo de nodos se eliminarán todos los nodos del grupo de nodos. Realice una copia de respaldo de los datos de manera oportuna para evitar la pérdida de datos.
- La eliminación de un nodo conducirá a la migración de pods, que puede afectar a los servicios. Realice esta operación durante las horas de menor actividad. Si los pods en el grupo de nodos tienen un selector de nodos específico y ninguno de los otros nodos en el clúster satisface el selector de nodos, los pods se volverán no programables.
- Al eliminar un grupo de nodos, el sistema establece todos los nodos del grupo de nodos actual en el estado no programado.

### Procedimiento

**Paso 1** Inicie sesión en la consola de CCE.

**Paso 2** Haga clic en el nombre del clúster y acceda a la consola del clúster. Elija **Nodos** en el panel de navegación y haga clic en la ficha **Node Pools** de la derecha.

**Paso 3** Elija **More > Delete** junto al nombre de un grupo de nodos para eliminar el grupo de nodos.

**Paso 4** Lea las precauciones en el cuadro de diálogo **Delete Node Pool**.

**Paso 5** En el cuadro de texto, haga clic en **Yes** para confirmar que desea continuar con la eliminación.

----**Fin**

# 5 Workloads

## 5.1 Overview

A workload is an application running on Kubernetes. No matter how many components are there in your workload, you can run it in a group of Kubernetes pods. A workload is an abstract model of a group of pods in Kubernetes. Workloads classified in Kubernetes include Deployments, StatefulSets, DaemonSets, jobs, and cron jobs.

CCE provides Kubernetes-native container deployment and management and supports lifecycle management of container workloads, including creation, configuration, monitoring, auto scaling, upgrade, uninstall, service discovery, and load balancing.

### Workload Lifecycle

**Tabla 5-1** Status description

Status	Description
Running	All pods are running.
Unready	A container is abnormal, the number of pods is 0, or the workload is in pending state.
Upgrading/Rolling back	The workload is being upgraded or rolled back.
Available	For a multi-pod Deployment, some pods are abnormal but at least one pod is available.
Completed	The task is successfully executed. This status is available only for common tasks.
Stopped	The workload is stopped and the number of pods changes to 0. This status is available for workloads earlier than v1.13.
Deleting	The workload is being deleted.
Pausing	The workload is being paused.

## 5.2 Creación de una carga de trabajo

### 5.2.1 Creación de una Deployment

#### Escenario

Las Deployment son cargas de trabajo (por ejemplo, Nginx) que no almacenan ningún dato o estado. Puede crear Deployment en la consola de CCE o ejecutando los comandos de kubectl.

#### Requisitos previos

- Antes de crear una carga de trabajo, debe tener un clúster disponible. Para obtener más información sobre cómo crear un clúster, consulte [Compra de un clúster de CCE](#).
- Para habilitar el acceso público a una carga de trabajo, asegúrese de que una EIP o un balanceador de carga se ha vinculado a al menos un nodo del clúster.

#### NOTA

Si un pod tiene contenedores múltiples, asegúrese de que los puertos utilizados por los contenedores no entren en conflicto entre sí. De lo contrario, la creación de la Deployment fallará.

#### Uso de la consola de CCE

**Paso 1** Inicie sesión en la consola de CCE.

**Paso 2** Haga clic en el nombre del clúster para ir a la consola del clúster, elija **Workloads** en el panel de navegación y haga clic en **Create Workload** en la esquina superior derecha.

**Paso 3** Establezca información básica sobre la carga de trabajo.

#### Basic Info

- **Workload Type:** Seleccione **Deployment**. Para obtener más información sobre los tipos de carga de trabajo, consulte [Overview](#).
- **Workload Name:** Introduzca el nombre de la carga de trabajo. Escriba de 1 a 63 caracteres que comienzan con una letra minúscula y terminan con una letra minúscula o un dígito. Solo se permiten letras minúsculas, dígitos y guiones (-).
- **Namespace:** Seleccione el espacio de nombres de la carga de trabajo. El valor predeterminado es **default**. También puede hacer clic en **Create Namespace** para crear uno. Para obtener más información, véase [Creación de un espacio de nombres](#).
- **Pods:** Ingrese el número de pods.
- **Container Runtime:** Un clúster de CCE utiliza runC de forma predeterminada, mientras que un clúster de CCE Turbo soporta runC y Kata. Para obtener más información sobre las diferencias, consulte [Contenedores de Kata y contenedores comunes](#).
- **Time Zone Synchronization:** Especifique si desea habilitar la sincronización de zona horaria. Una vez activada la sincronización de zona horaria, el contenedor y el nodo utilizan la misma zona horaria. La función de sincronización de zona horaria depende del disco local montado en el contenedor. No modifique ni elimine la zona horaria. Para obtener más información, véase [Configuración de la sincronización de zona horaria](#).

#### Configuración del contenedor

- Información del contenedor  
Se pueden configurar múltiples contenedores en un pod. Puede hacer clic en **Add Container** a la derecha para configurar varios contenedores para el pod.
  - **Basic Info:** Véase [Configuración de información básica del contenedor](#).
  - **Lifecycle:** Véase [Setting Container Lifecycle Parameters](#).
  - **Health Check:** Véase [Configuración de la comprobación de estado de un contenedor](#).
  - **Environment Variables:** Véase [Setting an Environment Variable](#).
  - **Data Storage:** Véase [Almacenamiento de contenedores](#).

#### NOTA

Si la carga de trabajo contiene más de un pod, los volúmenes de EVS no se pueden montar.

- **Security Context:** Establezca permisos de contenedor para proteger el sistema y otros contenedores de ser afectados. Introduzca el ID de usuario para establecer los permisos de contenedor y evitar que los sistemas y otros contenedores se vean afectados.
- **Logging:** Véase [Uso de ICAgent para recopilar logs de contenedores](#).
- **Image Access Credential:** Seleccione la credencial utilizada para acceder al repositorio de imágenes. El valor predeterminado es **default-secret**. Puede usar default-secret para acceder a las imágenes en SWR. Para obtener más información acerca de **default-secret**, consulte [default-secret](#).
- **GPU graphics card:** **All** está seleccionado de forma predeterminada. La instancia de carga de trabajo se programará en el nodo con el tipo de tarjeta gráfica de GPU especificado.

### Configuración de servicio

Se utiliza un Service para el acceso a pods. Con una dirección IP fija, un Service reenvía el tráfico de acceso a los pods y realiza el balanceo de carga para estos pods.

También puede crear un Service después de crear una carga de trabajo. Para obtener más información sobre el Service, consulte [Descripción general](#).

### Configuración avanzada

- **Upgrade:** Véase [Configuración de la política de actualización de carga de trabajo](#).
- **Scheduling:** Véase [Política de programación \(afinidad/antiafinidad\)](#).
- **Labels and Annotations:** Véase [Etiquetas y anotaciones de pod](#).
- **Toleration:** El uso de manchas y tolerancias permite (no a la fuerza) que el pod se programe en un nodo con las manchas correspondientes, y controla las políticas de desalojo del pod después de que el nodo donde se encuentra el pod esté contaminado. Para obtener más información, véase [Tolerancias](#).
- **DNS:** Véase [Configuración de DNS](#).
- **APM Settings:** Véase [Configuración de la configuración de APM para el análisis de cuello de botella del rendimiento](#).

**Paso 4** Haga clic en **Create Workload** en la esquina inferior derecha.

----Fin

## Uso de kubectl

El siguiente procedimiento utiliza Nginx como ejemplo para describir cómo crear una carga de trabajo con kubectl.

- Paso 1** Utilice kubectl para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).
- Paso 2** Cree y edite el archivo `nginx-deployment.yaml`. `nginx-deployment.yaml` es un nombre de archivo de ejemplo. Puede cambiar el nombre según sea necesario.

### vi nginx-deployment.yaml

El siguiente es un archivo YAML de ejemplo. Para obtener más información sobre las Deployments, consulte la [documentación de Kubernetes](#).

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - image: nginx # If you use an image from an open-source image registry,
          # enter the image name. If you use an image in My Images, obtain the image path
          # from SWR.
          imagePullPolicy: Always
          name: nginx
          imagePullSecrets:
            - name: default-secret
```

Para obtener más información sobre estos parámetros, consulte [Tabla 5-2](#).

**Tabla 5-2** Parámetro de YAML de Deployment

Parámetro	Descripción	Obligatorio/opcional
apiVersion	Versión de la API. <b>NOTA</b> Establezca este parámetro en función de la versión del clúster. <ul style="list-style-type: none"> <li>● Para los clústeres de v1.17 o posterior, el formato de apiVersion de las Deployments es <b>apps/v1</b>.</li> <li>● Para los clústeres de v1.15 o versiones anteriores, el formato de apiVersion de las Deployments es <b>extensions/v1beta1</b>.</li> </ul>	Obligatorio
kind	Tipo de un objeto creado.	Obligatorio
metadata	Metadatos de un objeto de recurso.	Obligatorio

Parámetro	Descripción	Obligatorio/opcional
name	Nombre de la Deployment.	Obligatorio
Spec	Descripción detallada de la Deployment.	Obligatorio
replicas	Número de pods.	Obligatorio
selector	Determina los pods de contenedor que puede gestionar la Deployment.	Obligatorio
strategy	Modo de actualización. Valores posibles: <ul style="list-style-type: none"> <li>● RollingUpdate</li> <li>● ReplaceUpdate</li> </ul> De forma predeterminada, se utiliza la actualización acumulativa.	Opcional
template	Descripción detallada de un pod de contenedor creado.	Obligatorio
metadata	Metadatos.	Obligatorio
labels	<b>metadata.labels:</b> Etiquetas de contenedor.	Opcional
spec: containers	<ul style="list-style-type: none"> <li>● <b>image</b> (obligatorio): Nombre de una imagen de contenedor.</li> <li>● <b>imagePullPolicy</b> (opcional): Política para la obtención de una imagen. Las opciones incluyen <b>Always</b> (intentando descargar imágenes cada vez), <b>Never</b> (solo usando imágenes locales) y <b>IfNotPresent</b> (utilizando imágenes locales si están disponibles; descargando imágenes si las imágenes locales no están disponibles). El valor predeterminado es <b>Always</b>.</li> <li>● <b>name</b> (obligatorio): Nombre del contenedor.</li> </ul>	Obligatorio
imagePullSecrets	Nombre del secreto utilizado durante la extracción de imágenes. Si se utiliza una imagen privada, este parámetro es obligatorio. <ul style="list-style-type: none"> <li>● Para extraer una imagen del Software Repository for Container (SWR), establezca este parámetro en <b>default-secret</b>.</li> <li>● Para extraer una imagen de un repositorio de imágenes de terceros, establezca este parámetro en el nombre del secreto creado.</li> </ul>	Opcional

**Paso 3** Cree una Deployment.

```
kubectl create -f nginx-deployment.yaml
```

Si se muestra la siguiente información, se está creando la Deployment.

```
deployment "nginx" created
```

**Paso 4** Consulte el estado de Deployment.

```
kubectl get deployment
```

Si se muestra la siguiente información, se está ejecutando la Deployment.

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
nginx	1/1	1	1	4m5s

#### Parameter description

- **NAME:** Nombre de la aplicación que se ejecuta en el pod.
- **READY:** indica el número de cargas de trabajo disponibles. El valor se muestra como "the number of available pods/the number of expected pods".
- **UP-TO-DATE:** indica el número de réplicas que se han actualizado.
- **AVAILABLE:** indica el número de pods disponibles.
- **AGE:** período en el que la Deployment sigue ejecutándose

**Paso 5** Si se accede a la Deployment con un Service de ClusterIP o de NodePort, agregue el Service correspondiente. Para obtener más información, véase [Red](#).

----Fin

## 5.2.2 Creación de un StatefulSet

### Escenario

StatefulSets es un tipo de cargas de trabajo cuyos datos o estado se almacenan mientras se ejecutan. Por ejemplo, MySQL es un StatefulSet porque necesita almacenar nuevos datos.

Un contenedor se puede migrar entre diferentes hosts, pero los datos no se almacenan en los hosts. Para almacenar datos de StatefulSet de forma persistente, conecte los volúmenes de almacenamiento de HA proporcionados por CCE a contenedor.

### Restricciones

- Al eliminar o ajustar un StatefulSet, el sistema no elimina los volúmenes de almacenamiento asociados con el StatefulSet para garantizar la seguridad de los datos.
- Cuando elimine un StatefulSet, reduzca el número de réplicas a **0** antes de eliminar el StatefulSet para que los pods del StatefulSet se puedan detener en orden.
- Cuando crea un StatefulSet, se requiere un Service sin cabeza para acceder a pods. Para obtener más información, véase [Headless Service](#).
- Cuando un nodo no está disponible, los pods se convierten en **Unready**. En este caso, debe eliminar manualmente los pods del StatefulSet para que los pods se puedan migrar a un nodo normal.

## Requisitos previos

- Antes de crear una carga de trabajo, debe tener un clúster disponible. Para obtener más información sobre cómo crear un clúster, consulte [Compra de un clúster de CCE](#).
- Para habilitar el acceso público a una carga de trabajo, asegúrese de que una EIP o un balanceador de carga se ha vinculado a al menos un nodo del clúster.

### NOTA

Si un pod tiene contenedores múltiples, asegúrese de que los puertos utilizados por los contenedores no entren en conflicto entre sí. De lo contrario, la creación del StatefulSet fallará.

## Uso de la consola de CCE

**Paso 1** Inicie sesión en la consola de CCE.

**Paso 2** Haga clic en el nombre del clúster para ir a la consola del clúster, elija **Workloads** en el panel de navegación y haga clic en **Create Workload** en la esquina superior derecha.

**Paso 3** Establezca información básica sobre la carga de trabajo.

### Informaciones básicas

- **Workload Type:** Seleccione **StatefulSet**. Para obtener más información sobre los tipos de carga de trabajo, consulte [Overview](#).
- **Workload Name:** Introduzca el nombre de la carga de trabajo. Escriba de 1 a 52 caracteres que comiencen con una letra minúscula y terminen con una letra o un dígito. Solo se permiten letras minúsculas, dígitos y guiones (-).
- **Namespace:** Seleccione el espacio de nombres de la carga de trabajo. El valor predeterminado es **default**. También puede hacer clic en **Create Namespace** para crear uno. Para obtener más información, véase [Creación de un espacio de nombres](#).
- **Pods:** Ingrese el número de pods.
- **Time Zone Synchronization:** Especifique si desea habilitar la sincronización de zona horaria. Una vez activada la sincronización de zona horaria, el contenedor y el nodo utilizan la misma zona horaria. La función de sincronización de zona horaria depende del disco local montado en el contenedor. No modifique ni elimine la zona horaria. Para obtener más información, véase [Configuración de la sincronización de zona horaria](#).

### Configuración del contenedor

- Información del contenedor

Se pueden configurar múltiples contenedores en un pod. Puede hacer clic en **Add Container** a la derecha para configurar varios contenedores para el pod.

- **Basic Info:** Véase [Configuración de información básica del contenedor](#).
- **Lifecycle:** Véase [Setting Container Lifecycle Parameters](#).
- **Health Check:** Véase [Configuración de la comprobación de estado de un contenedor](#).
- **Environment Variables:** Véase [Setting an Environment Variable](#).
- **Data Storage:** Véase [Almacenamiento de contenedores](#).

## 📖 NOTA

- **StatefulSets** admite los volúmenes de EVS aprovisionados dinámicamente.  
El montaje dinámico se consigue utilizando el campo **volumeClaimTemplates** y depende de la capacidad de creación dinámica de StorageClass. Un StatefulSet asocia cada pod con un PVC usando el campo **volumeClaimTemplates** y el PVC está unido al PV correspondiente. Por lo tanto, después de reprogramar el pod, los datos originales todavía se pueden montar basándose en el nombre de PVC.
- Después de crear una carga de trabajo, el almacenamiento que se monta dinámicamente no se puede actualizar.
- **Security Context:** Establezca permisos de contenedor para proteger el sistema y otros contenedores de ser afectados. Introduzca el ID de usuario para establecer los permisos de contenedor y evitar que los sistemas y otros contenedores se vean afectados.
- **Logging:** Véase [Uso de ICAgent para recopilar logs de contenedores](#).
- **Image Access Credential:** Seleccione la credencial utilizada para acceder al repositorio de imágenes. El valor predeterminado es **default-secret**. Puede usar **default-secret** para acceder a las imágenes en SWR. Para obtener más información acerca de **default-secret**, consulte [default-secret](#).
- **GPU graphics card:** **All** está seleccionado de forma predeterminada. La instancia de carga de trabajo se programará en el nodo con el tipo de tarjeta gráfica de GPU especificado.

### Parámetros de Service sin cabeza

Un Service sin cabeza se utiliza para resolver el problema de acceso mutuo entre pods en un StatefulSet. El Service sin cabeza proporciona un nombre de dominio de acceso fijo para cada pod. Para obtener más información, véase [Headless Service](#).

### Configuración de servicio

Se utiliza un Service para el acceso a pods. Con una dirección IP fija, un Service reenvía el tráfico de acceso a los pods y realiza el balanceo de carga para estos pods.

También puede crear un Service después de crear una carga de trabajo. Para obtener más información sobre el Service, consulte [Descripción general](#).

### Configuración avanzada

- **Upgrade:** Véase [Configuración de la política de actualización de carga de trabajo](#).
- **Scheduling:** Véase [Política de programación \(afinidad/antiafinidad\)](#).

- **Política de la gestión de instancias**

Para algunos sistemas distribuidos, la secuencia de StatefulSet es innecesaria y/o no debería ocurrir. Estos sistemas solo requieren unicidad e identificadores.

- **OrderedReady:** El StatefulSet desplegará, eliminará o escalará los pods en orden y uno por uno. (El StatefulSet continúa solo después de que el pod anterior esté listo o eliminado.) Esta es la política predeterminada.
- **Parallel:** El StatefulSet creará pods en paralelo para que coincidan con la escala deseada sin esperar, y eliminará todos los pods a la vez.
- **Toleration:** El uso de manchas y tolerancias permite (no a la fuerza) que el pod se programe en un nodo con las manchas correspondientes, y controla las políticas de desalojo del pod después de que el nodo donde se encuentra el pod esté contaminado. Para obtener más información, véase [Tolerancias](#).

- **Labels and Annotations:** Véase [Etiquetas y anotaciones de pod](#).
- **DNS:** Véase [Configuración de DNS](#).
- **APM Settings:** Véase [Configuración de la configuración de APM para el análisis de cuello de botella del rendimiento](#).

**Paso 4** Haga clic en **Create Workload** en la esquina inferior derecha.

----Fin

## Uso de kubectl

En este ejemplo, se utiliza una carga de trabajo nginx y el volumen de EVS se monta dinámicamente en él utilizando el campo `volumeClaimTemplates`.

**Paso 1** Utilice kubectl para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).

**Paso 2** Cree y edite el archivo `nginx-statefulset.yaml`.

`nginx-statefulset.yaml` es un nombre de archivo de ejemplo, y puede cambiarlo según sea necesario.

### vi nginx-statefulset.yaml

A continuación se proporciona un ejemplo del contenido del archivo. Para obtener más información sobre StatefulSet, consulte la [documentación de Kubernetes](#).

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: nginx
spec:
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: container-1
          image: nginx:latest
          imagePullPolicy: IfNotPresent
          resources:
            requests:
              cpu: 250m
              memory: 512Mi
            limits:
              cpu: 250m
              memory: 512Mi
          volumeMounts:
            - name: test
              readOnly: false
              mountPath: /usr/share/nginx/html
              subPath: ''
      imagePullSecrets:
        - name: default-secret
      dnsPolicy: ClusterFirst
      volumes: []
  serviceName: nginx-svc
  replicas: 2
  volumeClaimTemplates: # Dynamically mounts the EVS volume to the workload.
    - apiVersion: v1
```

```

kind: PersistentVolumeClaim
metadata:
  name: test
  namespace: default
  annotations:
    everest.io/disk-volume-type: SAS # SAS EVS volume type.
  labels:
    failure-domain.beta.kubernetes.io/region: ap-southeast-1 # region
    failure-domain.beta.kubernetes.io/zone: ap-southeast-1 # AZ where the
EVS volume is created. It must be the same as the AZ of the node.
spec:
  accessModes:
    - ReadWriteOnce # The value must be ReadWriteOnce for the EVS volume.
  resources:
    requests:
      storage: 10Gi
      storageClassName: csi-disk # Storage class name. The value is csi-disk
for the EVS volume.
  updateStrategy:
    type: RollingUpdate
    
```

### vi nginx-headless.yaml

```

apiVersion: v1
kind: Service
metadata:
  name: nginx-svc
  namespace: default
  labels:
    app: nginx
spec:
  selector:
    app: nginx
    version: v1
  clusterIP: None
  ports:
    - name: nginx
      targetPort: 80
      nodePort: 0
      port: 80
      protocol: TCP
  type: ClusterIP
    
```

**Paso 3** Cree una carga de trabajo y el servicio sin cabeza correspondiente.

### kubectl create -f nginx-statefulset.yaml

Si se muestra la siguiente información, el StatefulSet se ha creado correctamente.

```
statefulset.apps/nginx created
```

### kubectl create -f nginx-headless.yaml

Si se muestra la siguiente información, el servicio sin cabeza se ha creado correctamente.

```
service/nginx-svc created
```

**Paso 4** Si se accede a la carga de trabajo con un Service de ClusterIP o de NodePort, establezca el tipo de acceso de la carga de trabajo correspondiente. Para obtener más información, véase [Red](#).

----Fin

## 5.2.3 Creación de un DaemonSet

### Escenario

CCE proporciona capacidades de despliegue y gestión para múltiples tipos de contenedores y admite funciones de cargas de trabajo de contenedor como la creación, configuración, supervisión, ajuste, actualización, desinstalación, descubrimiento de servicios y balanceo de carga.

DaemonSet garantiza que solo se ejecute un pod en todos o algunos nodos. Cuando se agrega un nodo a un clúster, también se agrega un nuevo pod para el nodo. Cuando se elimina un nodo de un clúster, el pod también se recupera. Si se elimina un DaemonSet, se eliminarán todos los pods creados por él.

Los escenarios de aplicación típicos de un DaemonSet son los siguientes:

- Ejecute el daemon de almacenamiento de clúster, como glusterd o Ceph, en cada nodo.
- Ejecute el daemon de colección de logs, como Fluentd o Logstash, en cada nodo.
- Ejecute el daemon de supervisión, como Prometheus Node Exporter, collectd, Datadog agent, New Relic agent o Ganglia (gmond), en cada nodo.

Puede desplegar un DaemonSet para cada tipo de daemons en todos los nodos o desplegar varios DaemonSets para el mismo tipo de daemons. En el segundo caso, los DaemonSets tienen diferentes indicadores y diferentes requisitos en la memoria y la CPU para diferentes tipos de hardware.

### Requisitos previos

Antes de crear un DaemonSet debe tener un clúster disponible. Para obtener más información sobre cómo crear un clúster, consulte [Compra de un clúster de CCE](#).

### Uso de la consola de CCE

**Paso 1** Inicie sesión en la consola de CCE.

**Paso 2** Haga clic en el nombre del clúster para ir a la consola del clúster, elija **Workloads** en el panel de navegación y haga clic en **Create Workload** en la esquina superior derecha.

**Paso 3** Establezca información básica sobre la carga de trabajo.

#### Informaciones básicas

- **Workload Type:** Seleccione **DaemonSet**. Para obtener más información sobre los tipos de carga de trabajo, consulte [Overview](#).
- **Workload Name:** Introduzca el nombre de la carga de trabajo. Escriba de 1 a 63 caracteres que comienzan con una letra minúscula y terminan con una letra minúscula o un dígito. Solo se permiten letras minúsculas, dígitos y guiones (-).
- **Namespace:** Seleccione el espacio de nombres de la carga de trabajo. El valor predeterminado es **default**. También puede hacer clic en **Create Namespace** para crear uno. Para obtener más información, véase [Creación de un espacio de nombres](#).
- **Time Zone Synchronization:** Especifique si desea habilitar la sincronización de zona horaria. Una vez activada la sincronización de zona horaria, el contenedor y el nodo utilizan la misma zona horaria. La función de sincronización de zona horaria depende del disco local montado en el contenedor. No modifique ni elimine la zona horaria. Para obtener más información, véase [Configuración de la sincronización de zona horaria](#).

### Configuración del contenedor

- Información del contenedor

Se pueden configurar múltiples contenedores en un pod. Puede hacer clic en **Add Container** a la derecha para configurar varios contenedores para el pod.

- **Basic Info:** Véase [Configuración de información básica del contenedor](#).
- **Lifecycle:** Véase [Setting Container Lifecycle Parameters](#).
- **Health Check:** Véase [Configuración de la comprobación de estado de un contenedor](#).
- **Environment Variables:** Véase [Setting an Environment Variable](#).
- **Data Storage:** Véase [Almacenamiento de contenedores](#).

#### NOTA

Si la carga de trabajo contiene más de un pod, los volúmenes de EVS no se pueden montar.

- **Security Context:** Establezca permisos de contenedor para proteger el sistema y otros contenedores de ser afectados. Introduzca el ID de usuario para establecer los permisos de contenedor y evitar que los sistemas y otros contenedores se vean afectados.
- **Logging:** Véase [Uso de ICAgent para recopilar logs de contenedores](#).
- **Image Access Credential:** Seleccione la credencial utilizada para acceder al repositorio de imágenes. El valor predeterminado es **default-secret**. Puede usar default-secret para acceder a las imágenes en SWR. Para obtener más información acerca de **default-secret**, consulte [default-secret](#).
- **GPU graphics card:** **All** está seleccionado de forma predeterminada. La instancia de carga de trabajo se programará en el nodo con el tipo de tarjeta gráfica de GPU especificado.

### Configuración de servicio

Se utiliza un Service para el acceso a pods. Con una dirección IP fija, un Service reenvía el tráfico de acceso a los pods y realiza el balanceo de carga para estos pods.

También puede crear un Service después de crear una carga de trabajo. Para obtener más información sobre el Servicio, consulte [Descripción general](#).

### Configuración avanzada

- **Upgrade:** Véase [Configuración de la política de actualización de carga de trabajo](#).
- **Scheduling:** Véase [Política de programación \(afinidad/antiafinidad\)](#).
- **Labels and Annotations:** Véase [Etiquetas y anotaciones de pod](#).
- **Toleration:** El uso de manchas y tolerancias permite (no a la fuerza) que el pod se programe en un nodo con las manchas correspondientes, y controla las políticas de desalojo del pod después de que el nodo donde se encuentra el pod esté contaminado. Para obtener más información, véase [Tolerancias](#).
- **DNS:** Véase [Configuración de DNS](#).
- **APM Settings:** Véase [Configuración de la configuración de APM para el análisis de cuello de botella del rendimiento](#).

**Paso 4** Haga clic en **Create Workload** en la esquina inferior derecha.

----Fin

## Uso de kubectl

El siguiente procedimiento utiliza Nginx como ejemplo para describir cómo crear una carga de trabajo con kubectl.

**Paso 1** Utilice kubectl para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).

**Paso 2** Cree y edite el archivo `nginx-daemonset.yaml`. `nginx-daemonset.yaml` es un nombre de archivo de ejemplo, y puede cambiarlo según sea necesario.

### vi nginx-daemonset.yaml

El contenido del archivo de descripción es el siguiente: A continuación se proporciona un ejemplo. Para obtener más información sobre DaemonSets, consulte [Documentos de Kubernetes](#).

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: nginx-daemonset
  labels:
    app: nginx-daemonset
spec:
  selector:
    matchLabels:
      app: nginx-daemonset
  template:
    metadata:
      labels:
        app: nginx-daemonset
    spec:
      nodeSelector:
        # Node selection. A pod is created on a node
        # only when the node meets daemon=need.
        daemon: need
      containers:
      - name: nginx-daemonset
        image: nginx:alpine
        resources:
          limits:
            cpu: 250m
            memory: 512Mi
          requests:
            cpu: 250m
            memory: 512Mi
        imagePullSecrets:
        - name: default-secret
```

El parámetro **replicas** usado para definir una Deployment o StatefulSet no existe en la configuración anterior para un DaemonSet porque cada nodo tiene solo una réplica. Está arreglado.

El nodeSelector de la plantilla de pod anterior especifica que un pod solo se crea en los nodos que cumplen con **daemon=need** como se muestra en la siguiente figura. Si desea crear un pod en cada nodo, elimine la etiqueta.

**Paso 3** Cree un DaemonSet.

### **kubectl create -f nginx-daemonset.yaml**

Si se muestra la siguiente información, se está creando el DaemonSet.

```
daemonset.apps/nginx-daemonset created
```

**Paso 4** Consulte el estado DaemonSet.

### kubectl get ds

```
$ kubectl get ds
NAME                DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE
SELECTOR    AGE
nginx-daemonset    1         1         0       1             0
daemon=need    116s
```

**Paso 5** Si se accede a la carga de trabajo con un Service de ClusterIP o de NodePort, establezca el tipo de acceso de la carga de trabajo correspondiente. Para obtener más información, véase [Red](#).

----Fin

## 5.2.4 Creación de un trabajo

### Escenario

Los trabajos son de corta duración y se ejecutan durante un cierto tiempo hasta su finalización. Pueden ejecutarse inmediatamente después de ser desplegados. Se completa después de que sale normalmente (salida 0).

Un trabajo es un objeto de recurso que se utiliza para controlar tareas por lotes. Es diferente de una carga de trabajo de servo a largo plazo (como Deployment y StatefulSet).

Un trabajo se inicia y finaliza en los momentos específicos, mientras que una carga de trabajo de servo a largo plazo se ejecuta sin cesar a menos que se termine. Los pods gestionados por un trabajo se cierran automáticamente después de completar correctamente el trabajo en función de las configuraciones de usuario. El indicador de éxito varía según la política `spec.completions`.

- Trabajos únicos: Un solo pod se ejecuta una vez hasta que se termina correctamente.
- Trabajos con un recuento de éxito fijo: N pods se ejecutan hasta que se termina correctamente.
- Un trabajo de cola se considera completado en función del éxito global confirmado por la aplicación.

### Requisitos previos

Se han creado recursos. Para obtener más información, véase [Creación de un nodo](#). Si hay clústeres y nodos disponibles, no es necesario volver a crearlos.

### Uso de la consola de CCE

**Paso 1** Inicie sesión en la consola de CCE.

**Paso 2** Haga clic en el nombre del clúster para ir a la consola del clúster, elija **Workloads** en el panel de navegación y haga clic en **Create Workload** en la esquina superior derecha.

**Paso 3** Establezca información básica sobre la carga de trabajo.

#### Informaciones básicas

- **Workload Type:** Seleccione **Job**. Para obtener más información sobre los tipos de carga de trabajo, consulte [Overview](#).

- **Workload Name:** Introduzca el nombre de la carga de trabajo. Escriba de 1 a 63 caracteres que comienzan con una letra minúscula y terminan con una letra minúscula o un dígito. Solo se permiten letras minúsculas, dígitos y guiones (-).
- **Namespace:** Seleccione el espacio de nombres de la carga de trabajo. El valor predeterminado es **default**. También puede hacer clic en **Create Namespace** para crear uno. Para obtener más información, véase [Creación de un espacio de nombres](#).
- **Pods:** Ingrese el número de pods.

#### Configuración del contenedor

- Información del contenedor

Se pueden configurar múltiples contenedores en un pod. Puede hacer clic en **Add Container** a la derecha para configurar varios contenedores para el pod.

- **Basic Info:** Véase [Configuración de información básica del contenedor](#).
- **Lifecycle:** Véase [Setting Container Lifecycle Parameters](#).
- **Environment Variables:** Véase [Setting an Environment Variable](#).
- **Data Storage:** Véase [Almacenamiento de contenedores](#).

#### NOTA

Si la carga de trabajo contiene más de un pod, los volúmenes de EVS no se pueden montar.

- **Logging:** Véase [Uso de ICAgent para recopilar logs de contenedores](#).
- **Image Access Credential:** Seleccione la credencial utilizada para acceder al repositorio de imágenes. El valor predeterminado es **default-secret**. Puede usar default-secret para acceder a las imágenes en SWR. Para obtener más información acerca de **default-secret**, consulte [default-secret](#).
- **GPU graphics card:** **All** está seleccionado de forma predeterminada. La instancia de carga de trabajo se programará en el nodo con el tipo de tarjeta gráfica de GPU especificado.

#### Configuración avanzada

- **Labels and Annotations:** Véase [Etiquetas y anotaciones de pod](#).
- **Configuración de trabajo**
  - **Parallel Pods:** Número máximo de pods que pueden ejecutarse en paralelo durante la ejecución del trabajo. El valor no puede ser mayor que el número total de pods del trabajo.
  - **Timeout (s):** Una vez que un trabajo llega a este momento, el estado del trabajo falla y se eliminarán todos los pods de este trabajo. Si deja este parámetro en blanco, el trabajo nunca se agotará.

**Paso 4** Haga clic en **Create Workload** en la esquina inferior derecha.

----Fin

## Uso de kubectl

Un trabajo tiene los siguientes parámetros de configuración:

- **spec.template:** tiene el mismo esquema que un pod.
- **RestartPolicy:** solo se puede establecer en **Never** o **OnFailure**.
- Para un trabajo de un solo pod, el trabajo finaliza después de que el pod se ejecute correctamente de forma predeterminada.

- **.spec.completions**: indica el número de pods que deben ejecutarse correctamente para finalizar un trabajo. El valor predeterminado es **1**.
- **.spec.parallelism**: indica el número de pods que se ejecutan simultáneamente. El valor predeterminado es **1**.
- **spec.backoffLimit**: indica el número máximo de reintentos realizados si un pod falla. Cuando se alcanza el límite, el pod no volverá a intentarlo.
- **.spec.activeDeadlineSeconds**: indica el tiempo de funcionamiento de los pods. Una vez alcanzado el tiempo, se terminan todos los pods del trabajo. La prioridad de **.spec.activeDeadlineSeconds** es mayor que la de **.spec.backoffLimit**. Es decir, si un trabajo alcanza **.spec.activeDeadlineSeconds**, el **spec.backoffLimit** se omite.

Según la configuración de **.spec.completions** y de **.spec.Parallelism**, los trabajos se clasifican en los siguientes tipos.

**Tabla 5-3** Tipos de trabajo

Tipo de trabajo	Descripción	Ejemplo
Trabajos únicos	Un solo pod funciona una vez hasta que se termina correctamente.	Migración de bases de datos
Trabajos con un recuento de finalización fijo	Un pod se ejecuta hasta alcanzar el conteo de <b>completions</b> especificado.	Pod de procesamiento de colas de trabajo
Trabajos paralelos con un recuento de finalización fijo	Se ejecutan varios pods hasta alcanzar el recuento de <b>completions</b> especificado.	Múltiples pods para procesar colas de trabajo simultáneamente
Trabajos paralelos	Uno o más pods se ejecutan hasta que se termina correctamente.	Múltiples pods para procesar colas de trabajo simultáneamente

El siguiente es un trabajo de ejemplo, que calcula  $\pi$  hasta 2000º dígito e imprime la salida.

```
apiVersion: batch/v1
kind: Job
metadata:
  name: myjob
spec:
  completions: 50          # 50 pods need to be run to finish a job. In this
                           # example,  $\pi$  is printed for 50 times.
  parallelism: 5           # 5 pods are run in parallel.
  backoffLimit: 5         # The maximum number of retry times is 5.
  template:
    spec:
      containers:
      - name: pi
        image: perl
        command: ["perl", "-Mbignum=bpi", "-wle", "print bpi(2000)"]
        restartPolicy: Never
```

### Descripción

- **apiVersion: batch/v1** indica la versión del trabajo actual.
- **kind: Job** indica que el recurso actual es un trabajo.
- **restartPolicy: Never** indica la política de reinicio actual. Para los trabajos, este parámetro solo se puede establecer en **Never** o **OnFailure**. Para otros controladores (por ejemplo, Deployments), puede establecer este parámetro en **Always**.

### Ejecutar el trabajo.

#### Paso 1 Comience el trabajo.

```
[root@k8s-master k8s]# kubectl apply -f myjob.yaml
job.batch/myjob created
```

#### Paso 2 Vea los detalles del trabajo.

##### kubectl get job

```
[root@k8s-master k8s]# kubectl get job
NAME          COMPLETIONS  DURATION  AGE
myjob         50/50         23s       3m45s
```

Si el valor de **COMPLETIONS** es **50/50**, el trabajo se ejecuta correctamente.

#### Paso 3 Consulte el estado del pod.

##### kubectl get pod

```
[root@k8s-master k8s]# kubectl get pod
NAME          READY  STATUS    RESTARTS  AGE
myjob-29qlw   0/1    Completed  0          4m5s
...
```

Si el estado es **Completed**, el trabajo se ha completado.

#### Paso 4 Vea los registros de pod.

##### kubectl logs

```
# kubectl logs myjob-29qlw
3.14159265358979323846264338327950288419716939937510582097494459230781640628620899
8628034825342117067982148086513282306647093844609550582231725359408128481117450284
1027019385211055596446229489549303819644288109756659334461284756482337867831652712
0190914564856692346034861045432664821339360726024914127372458700660631558817488152
0920962829254091715364367892590360011330530548820466521384146951941511609433057270
3657595919530921861173819326117931051185480744623799627495673518857527248912279381
8301194912983367336244065664308602139494639522473719070217986094370277053921717629
3176752384674818467669405132000568127145263560827785771342757789609173637178721468
4409012249534301465495853710507922796892589235420199561121290219608640344181598136
2977477130996051870721134999999837297804995105973173281609631859502445945534690830
2642522308253344685035261931188171010003137838752886587533208381420617177669147303
5982534904287554687311595628638823537875937519577818577805321712268066130019278766
1119590921642019893809525720106548586327886593615338182796823030195203530185296899
5773622599413891249721775283479131515574857242454150695950829533116861727855889075
0983817546374649393192550604009277016711390098488240128583616035637076601047101819
4295559619894676783744944825537977472684710404753464620804668425906949129331367702
8989152104752162056966024058038150193511253382430035587640247496473263914199272604
2699227967823547816360093417216412199245863150302861829745557067498385054945885869
2699569092721079750930295532116534498720275596023648066549911988183479775356636980
7426542527862551818417574672890977772793800081647060016145249192173217214772350141
4419735685481613611573525521334757418494684385233239073941433345477624168625189835
694856209921922218427255025425688767179049460165346680498862723279178608578438382
7967976681454100953883786360950680064225125205117392984896084128488626945604241965
2850222106611863067442786220391949450471237137869609563643719172874677646575739624
138908658326459958133904780275901
```

----Fin

## Operaciones relacionadas

Una vez creado un trabajo único, puede realizar las operaciones que aparecen en la lista de [Tabla 5-4](#).

**Tabla 5-4** Otras operaciones

Operación	Descripción
Edición de un archivo YAML	Haga clic en <b>More &gt; Edit YAML</b> junto al nombre del trabajo para editar el archivo YAML correspondiente al trabajo actual.
Eliminación de un trabajo	<ol style="list-style-type: none"> <li>1. Seleccione el trabajo que desea eliminar y haga clic en <b>Delete</b> en la columna <b>Operation</b>.</li> <li>2. Haga clic en <b>Yes</b>.                      Los trabajos eliminados no se pueden restaurar. Tenga cuidado al eliminar un trabajo.</li> </ol>

## 5.2.5 Creación de un trabajo de cron

### Escenario

Un trabajo cron se ejecuta en una programación repetida. Puede realizar la sincronización de tiempo para todos los nodos activos en un punto de tiempo fijo.

Un trabajo de cron se ejecuta periódicamente a la hora especificada. Es similar al crontab de Linux. Un trabajo de cron tiene las siguientes características:

- Solo se ejecuta una vez a la hora especificada.
- Se ejecuta periódicamente a la hora especificada.

El uso típico de un trabajo de cron es el siguiente:

- Programa los trabajos a la hora especificada.
- Crea trabajos para ejecutarse periódicamente, por ejemplo, copia de respaldo de la base de datos y envío de correo electrónico.

### Requisitos previos

Se han creado los recursos. Para obtener más información, véase [Creación de un nodo](#).

### Uso de la consola de CCE

**Paso 1** Inicie sesión en la consola de CCE.

**Paso 2** Haga clic en el nombre del clúster para ir a la consola del clúster, elija **Workloads** en el panel de navegación y haga clic en **Create Workload** en la esquina superior derecha.

**Paso 3** Establezca información básica sobre la carga de trabajo.

#### Informaciones básicas

- **Workload Type:** Seleccione **Cron Job**. Para obtener más información sobre los tipos de carga de trabajo, consulte [Overview](#).

- **Workload Name:** Introduzca el nombre de la carga de trabajo. Escriba de 1 a 52 caracteres que comiencen con una letra minúscula y terminen con una letra o un dígito. Solo se permiten letras minúsculas, dígitos y guiones (-).
- **Namespace:** Seleccione el espacio de nombres de la carga de trabajo. El valor predeterminado es **default**. También puede hacer clic en **Create Namespace** para crear uno. Para obtener más información, véase [Creación de un espacio de nombres](#).

### Configuración del contenedor

- Información del contenedor  
Se pueden configurar múltiples contenedores en un pod. Puede hacer clic en **Add Container** a la derecha para configurar varios contenedores para el pod.
  - **Basic Info:** Véase [Configuración de información básica del contenedor](#).
  - **Lifecycle:** Véase [Setting Container Lifecycle Parameters](#).
  - **Environment Variables:** Véase [Setting an Environment Variable](#).
- **Image Access Credential:** Seleccione la credencial utilizada para acceder al repositorio de imágenes. El valor predeterminado es **default-secret**. Puede usar default-secret para acceder a las imágenes en SWR. Para obtener más información acerca de **default-secret**, consulte [default-secret](#).
- **GPU graphics card:** **All** está seleccionado de forma predeterminada. La instancia de carga de trabajo se programará en el nodo con el tipo de tarjeta gráfica de GPU especificado.

### Horarios

- **Concurrency Policy:** Se admiten los tres modos siguientes:
  - **Forbid:** No se puede crear un nuevo trabajo antes de completar el trabajo anterior.
  - **Allow:** El trabajo de cron permite trabajos en ejecución simultánea, que se adelantan a los recursos del clúster.
  - **Replace:** Un nuevo trabajo sustituye al anterior cuando es el momento de crear un trabajo pero el anterior no se ha completado.
- **Policy Settings:** especifica cuándo se ejecuta un nuevo trabajo de cron. La configuración de política en YAML se implementa con las expresiones de cron.
  - Un trabajo de cron se ejecuta en un intervalo fijo. La unidad puede ser minuto, hora, día o mes. Por ejemplo, si se ejecuta un trabajo de cron cada 30 minutos y la expresión de cron correspondiente es `*/30 * * * *`, el tiempo de ejecución comienza de 0 en el rango de unidades, por ejemplo, `00:00`, `00:30:00`, `01:00:00`, y ...
  - El trabajo de cron se ejecuta a una hora fija (por mes). Por ejemplo, si se ejecuta un trabajo de cron a las 00:00 del primer día de cada mes, la expresión cron será `0 0 1 */1 *` y el tiempo de ejecución será `****-01-01 00:00:00`, `****-02-01 00:00:00` y ...
  - El trabajo de cron se ejecuta por semana. Por ejemplo, si se ejecuta un trabajo de cron a las 00:00 todos los lunes, la expresión cron será `0 0 * * 1` y el tiempo de ejecución será `****_**-01 00:00:00 on Monday`, `****_**-08 00:00:00 on Monday` y ...
  - **Custom Cron Expression:** Para obtener detalles sobre cómo usar expresiones de cron, consulte [CRON](#).

 **NOTA**

- Si un trabajo de cron se ejecuta a una hora fija (por mes) y el número de días en un mes no existe, el trabajo de cron no se ejecutará en este mes. Por ejemplo, la ejecución omitirá febrero si la fecha está establecida en 30.
- Debido a la definición de cron, el período fijo no es un período estricto. El intervalo de unidades de tiempo se divide de 0 por período. Por ejemplo, si la unidad es minuto, el valor varía de 0 a 59. Si el valor no se puede dividir exactamente, se restablece el último período. Por lo tanto, un período exacto puede representarse solo cuando el período puede dividirse uniformemente.

Tome un trabajo de cron que se ejecuta por hora como ejemplo. Como **/2, /3, /4, /6, /8 y /12** puede dividir exactamente 24 horas, se puede representar un período preciso. Si se utiliza otro período, el último período se restablecerá al comienzo de un nuevo día. Por ejemplo, si la expresión de cron es **\* \*/12 \* \* \***, el tiempo de ejecución es **00:00:00** y **12:00:00** todos los días. Si la expresión de cron es **\* \*/13 \* \* \***, el tiempo de ejecución es **00:00:00** y **13:00:00** todos los días. A las 00:00 del día siguiente, el tiempo de ejecución se actualiza incluso si el período no alcanza las 13 horas.

- **Job Records:** Puede establecer el número de trabajos que se ejecutan correctamente o que no se ejecutan. Establezca un límite en **0** corresponde a mantener ninguno de los trabajos después de que terminen.

**Configuración avanzada**

- **Labels and Annotations:** Véase [Etiquetas y anotaciones de pod](#).

**Paso 4** Haga clic en **Create Workload** en la esquina inferior derecha.

----Fin

## Uso de kubectl

Un trabajo de cron tiene los siguientes parámetros de configuración:

- **.spec.schedule:** toma una cadena de formato **Cron**, por ejemplo, **0 \* \* \* \*** o **@hourly**, como el horario de los trabajos a crear y ejecutar.
- **.spec.jobTemplate:** especifica los trabajos que se van a ejecutar y tiene el mismo esquema que cuando se está [Creando un trabajo con kubectl](#).
- **.spec.startingDeadlineSeconds:** especifica la fecha límite para iniciar un trabajo.
- **.spec.concurrencyPolicy:** especifica cómo tratar las ejecuciones simultáneas de un trabajo creado por el trabajo de Cron. The following options are supported:
  - **Allow** (valor predeterminado): permite ejecutar trabajos simultáneamente.
  - **Forbid:** prohíbe las ejecuciones simultáneas, omitiendo la siguiente ejecución si la anterior aún no ha terminado.
  - **Replace:** cancela el trabajo en ejecución y lo reemplaza por uno nuevo.

A continuación se muestra un ejemplo de trabajo de cron, que se guarda en el archivo **cronjob.yaml**.

```
apiVersion: batch/v1beta1
kind: CronJob
metadata:
  name: hello
spec:
  schedule: "*/1 * * * *"
  jobTemplate:
    spec:
      template:
```

```
spec:
  containers:
  - name: hello
    image: busybox
    args:
    - /bin/sh
    - -c
    - date; echo Hello from the Kubernetes cluster
  restartPolicy: OnFailure
```

**Ejecute el trabajo.**

**Paso 1** Cree un trabajo de cron.

**kubectl create -f cronjob.yaml**

La información que aparecerá en pantalla será similar a la información siguiente:

```
cronjob.batch/hello created
```

**Paso 2** Consulte el estado de ejecución del trabajo de cron:

**kubectl get cronjob**

NAME	SCHEDULE	SUSPEND	ACTIVE	LAST SCHEDULE	AGE
hello	*/* * * * *	False	0	<none>	9s

**kubectl get jobs**

NAME	COMPLETIONS	DURATION	AGE
hello-1597387980	1/1	27s	45s

**kubectl get pod**

NAME	READY	STATUS	RESTARTS	AGE
hello-1597387980-tjv8f	0/1	Completed	0	114s
hello-1597388040-lckg9	0/1	Completed	0	39s

**kubectl logs hello-1597387980-tjv8f**

```
Fri Aug 14 06:56:31 UTC 2020
Hello from the Kubernetes cluster
```

**kubectl delete cronjob hello**

```
cronjob.batch "hello" deleted
```

**AVISO**

Cuando se elimina un trabajo cron, también se eliminan los trabajos y pods relacionados.

----Fin

**Operaciones relacionadas**

Después de crear un trabajo de cron, puede realizar las operaciones que aparecen en la lista de [Tabla 5-5](#).

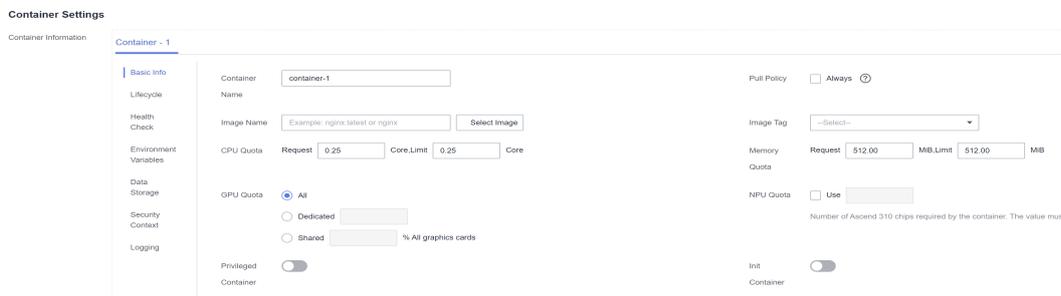
**Tabla 5-5** Otras operaciones

Operación	Descripción
Edición de un archivo YAML	Haga clic en <b>More &gt; Edit YAML</b> junto al nombre del trabajo de cron para editar el archivo YAML del trabajo actual.
Detener un trabajo de cron	<ol style="list-style-type: none"> <li>1. Seleccione el trabajo que desea detener y haga clic en <b>Stop</b> en la columna <b>Operation</b>.</li> <li>2. Haga clic en <b>Yes</b>.</li> </ol>
Eliminación de un trabajo de cron	<ol style="list-style-type: none"> <li>1. Seleccione el trabajo de cron que desea eliminar y haga clic en <b>More &gt; Delete</b> en la columna <b>Operation</b>.</li> <li>2. Haga clic en <b>Yes</b>.                      Los trabajos eliminados no se pueden restaurar. Por lo tanto, tenga cuidado al eliminar un trabajo.</li> </ol>

## 5.3 Configuring a Container

### 5.3.1 Configuración de información básica del contenedor

Una carga de trabajo es un modelo abstracto de un grupo de pods. Un pod puede encapsular uno o más contenedores. Puede hacer clic en **Add Container** en la esquina superior derecha para agregar varias imágenes de contenedor y configurarlas por separado.



**Tabla 5-6** Parámetros de imagen

Parámetro	Descripción
Container Name	Denomine el contenedor.
Image Name	Haga clic en <b>Select Image</b> y seleccione la imagen utilizada por el contenedor. Si necesita utilizar una imagen de terceros, consulte <a href="#">Uso de una imagen de terceros</a> .
Image Tag	Seleccione la etiqueta de imagen que se va a desplegar.

Parámetro	Descripción
Pull Policy	Política de actualización o extracción de imágenes. Si selecciona <b>Always</b> , cada vez se extrae la imagen del repositorio de imágenes. Si no selecciona <b>Always</b> , se utilizará preferentemente la imagen existente del nodo. Si la imagen no existe, la imagen se extrae del repositorio de imágenes.
CPU Quota	<ul style="list-style-type: none"> <li>● <b>Request</b>: número mínimo de núcleos de CPU requeridos por un contenedor. El valor predeterminado es 0.25 núcleos.</li> <li>● <b>Limit</b>: número máximo de núcleos de CPU disponibles para un contenedor. No deje <b>Limit</b> sin especificar. De lo contrario, se producirá un uso intensivo de los recursos de contenedor y su carga de trabajo puede mostrar un comportamiento inesperado.</li> </ul>
Memory Quota	<ul style="list-style-type: none"> <li>● <b>Request</b>: cantidad mínima de memoria requerida por un contenedor. El valor predeterminado es 512 MiB.</li> <li>● <b>Limit</b>: cantidad máxima de memoria disponible para un contenedor. Cuando el uso de memoria excede el límite de memoria especificado, el contenedor se terminará.</li> </ul> <p>Para obtener más información acerca de <b>Request</b> y <b>Limit</b>, consulte <a href="#">Establecimiento de las especificaciones del contenedor</a>.</p>
GPU Quota	<p>Solo se puede configurar cuando el clúster contiene los nodos de GPU.</p> <ul style="list-style-type: none"> <li>● <b>All</b>: no se utiliza la GPU.</li> <li>● <b>Dedicated</b>: los recursos de la GPU son utilizados exclusivamente por el contenedor.</li> <li>● <b>Shared</b>: porcentaje de recursos de GPU utilizados por el contenedor. Por ejemplo, si este parámetro se establece en <b>10%</b>, el contenedor utiliza el 10% de los recursos de la GPU.</li> </ul>
NPU Quota	<p>Número de fichas Ascend 310 requeridas por el contenedor. El valor debe ser un entero.</p> <p>Este parámetro solo se puede establecer después de instalar el complemento npu.</p>
Privileged Container	<p>Los programas en un contenedor privilegiado tienen ciertos privilegios.</p> <p>Si <b>Privileged Container</b> está habilitado, se asignan privilegios al contenedor. Por ejemplo, los contenedores privilegiados pueden manipular dispositivos de red en la máquina host y modificar los parámetros del núcleo.</p>
Init Container	<p>Indica si se debe utilizar el contenedor como contenedor de inicio.</p> <p>Un contenedor de inicio es un contenedor especial que se ejecuta antes que los contenedores de la aplicación en un pod. Para obtener más información, consulte <a href="#">Init Container</a>.</p>

## 5.3.2 Uso de una imagen de terceros

### Escenario

CCE le permite crear cargas de trabajo utilizando imágenes extraídas de repositorios de imágenes de terceros.

Generalmente, se puede acceder a un repositorio de imágenes de terceros solo después de la autenticación (usando su cuenta y contraseña). CCE utiliza la autenticación basada en secreto para extraer imágenes. Por lo tanto, debe crear un secreto para un repositorio de imágenes antes de extraer imágenes del repositorio.

### Requisitos previos

El nodo donde se ejecuta la carga de trabajo es accesible desde las redes públicas.

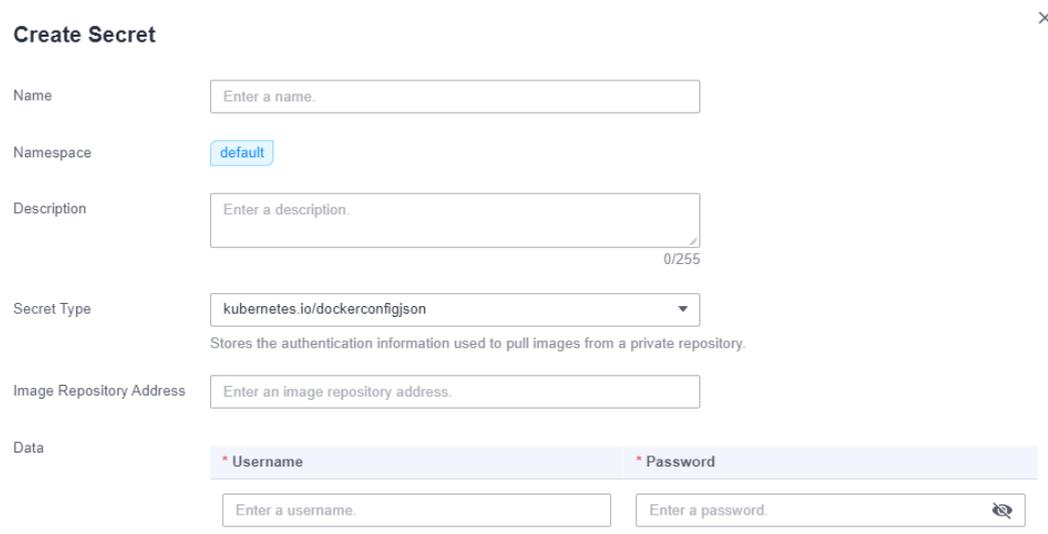
### Uso de la consola

**Paso 1** Cree un secreto para acceder a un repositorio de imágenes de terceros.

Haga clic en el nombre del clúster y acceda a la consola del clúster. En el panel de navegación, elija **ConfigMaps and Secrets**. En la página de la ficha **Secrets**, haga clic en **Create Secret** en la esquina superior derecha. Ajusta **Secret Type** a **kubernetes.io/dockerconfigjson**. Para obtener más información, véase [Creación de un secreto](#).

Introduzca el nombre de usuario y la contraseña utilizados para acceder al repositorio de imágenes de terceros.

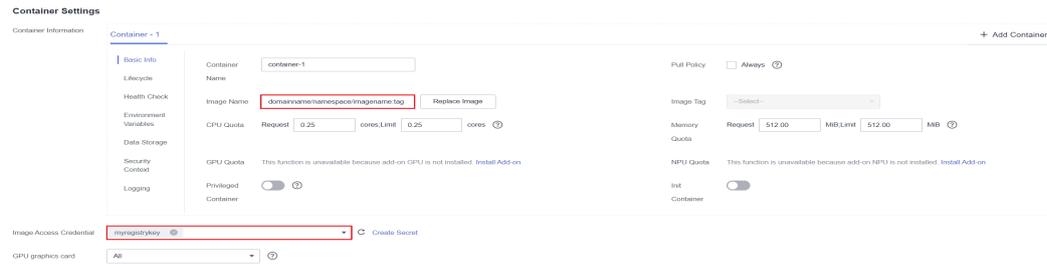
**Figura 5-1** Creación de un secreto



The screenshot shows the 'Create Secret' form with the following fields and values:

- Name:** Enter a name.
- Namespace:** default
- Description:** Enter a description. (0/255 characters)
- Secret Type:** kubernetes.io/dockerconfigjson (Dropdown menu)
- Image Repository Address:** Enter an image repository address.
- Data:** A table with two columns: **\* Username** and **\* Password**. Both fields have input boxes with placeholder text 'Enter a username.' and 'Enter a password.' respectively.

**Paso 2** Al crear una carga de trabajo, puede introducir una ruta de imagen privada en el formato **domainname/namespace/imagenname:tag** en **Image Name** y seleccionar la clave creada.



**Paso 3** Defina otros parámetros y haga clic en **Create Workload**.

----Fin

## Uso de kubectl

**Paso 1** Utilice kubectl para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).

**Paso 2** Cree un secreto del tipo dockercfg usando kubectl.

```
kubectl create secret docker-registry myregistrykey --docker-
server=DOCKER_REGISTRY_SERVER --docker-username=DOCKER_USER --docker-
password=DOCKER_PASSWORD --docker-email=DOCKER_EMAIL
```

En los comandos anteriores, **myregistrykey** indica el nombre de secreto y otros parámetros se describen de la siguiente manera:

- **DOCKER\_REGISTRY\_SERVER**: dirección de un repositorio de imágenes de terceros, por ejemplo, **www.3rdregistry.com** o **10.10.10:443**
- **DOCKER\_USER**: cuenta utilizada para iniciar sesión en un repositorio de imágenes de terceros
- **DOCKER\_PASSWORD**: contraseña utilizada para iniciar sesión en un repositorio de imágenes de terceros
- **DOCKER\_EMAIL**: correo electrónico de un repositorio de imágenes de terceros

**Paso 3** Utilice una imagen de terceros para crear una carga de trabajo.

Un secreto dockercfg se utiliza para la autenticación cuando se obtiene una imagen privada. A continuación se muestra un ejemplo de uso de myregistrykey para la autenticación.

```
apiVersion: v1
kind: Pod
metadata:
  name: foo
  namespace: default
spec:
  containers:
    - name: foo
      image: www.3rdregistry.com/janedoe/awesomeapp:v1
  imagePullSecrets:
    - name: myregistrykey #Use the created secret.
```

----Fin

## 5.3.3 Establecimiento de las especificaciones del contenedor

### Escenario

CCE le permite establecer límites de recursos para contenedores agregados durante la creación de cargas de trabajo. Puede solicitar y limitar las cuotas de CPU y memoria utilizadas por cada pod en una carga de trabajo.

### Significados

Para **CPU** y **Memory**, los significados de **Request** y **Limit** son los siguientes:

- **Request:** El sistema programa un pod para el nodo que cumple con los requisitos para el despliegue de carga de trabajo en función del valor de solicitud.
- **Limit:** El sistema limita los recursos utilizados por la carga de trabajo en función del valor límite.

Si un nodo tiene suficientes recursos, el pod de este nodo puede usar más recursos de los solicitados, pero no más que limitados.

Por ejemplo, si establece la solicitud de memoria de un contenedor en 1 GiB y el valor límite en 2 GiB, un pod se programa en un nodo con CPU de 8 GiB sin ningún otro pod en ejecución. En este caso, el pod puede usar más de 1 GiB de memoria cuando la carga es pesada, pero el uso de memoria no puede exceder los 2 GiB. Si un proceso en un contenedor intenta usar los recursos más de 2 GiB, el núcleo del sistema intenta terminar el proceso. Como resultado, se produce un error de OOM.

#### NOTA

Al crear una carga de trabajo, se recomienda establecer los límites superior e inferior de los recursos de CPU y memoria. Si no se establecen los límites de recursos superior e inferior para una carga de trabajo, una pérdida de recursos de esta carga de trabajo hará que los recursos no estén disponibles para otras cargas de trabajo desplegadas en el mismo nodo. Además, las cargas de trabajo que no tienen límites de recursos superiores e inferiores no se pueden supervisar con precisión.

### Descripción de configuración

En los servicios de producción reales, la relación recomendada de **Request** a **Limit** es de aproximadamente 1:1.5. Para algunos servicios sensibles, la proporción recomendada es 1:1. Si el **Request** es demasiado pequeño y el **Limit** es demasiado grande, los recursos de nodo se comprometen en exceso. Durante los picos de servicio, la memoria o la CPU de un nodo se puede utilizar. Como resultado, el nodo no está disponible.

- Cuotas de CPU:

**Tabla 5-7** Descripción de las cuotas de CPU

Parámetro	Descripción
CPU request	Número mínimo de núcleos de CPU requeridos por un contenedor. Los recursos se programan para el contenedor en función de este valor. El contenedor se puede programar para este nodo solo cuando el total de CPU disponible en el nodo es mayor o igual que el número de aplicaciones de CPU en contenedores.

Parámetro	Descripción
CPU limit	Número máximo de núcleos de CPU disponibles para un contenedor.

### Configuración recomendada

CPU real disponible de un nodo  $\geq$  Suma de los límites de CPU de todos los contenedores en el nodo actual  $\geq$  Suma de las solicitudes de CPU de todos los contenedores en el nodo actual. Puede ver las CPU disponibles reales de un nodo en la consola de CCE (**Resource Management > Nodes > Allocatable**).

- Cuotas de memoria:

**Tabla 5-8** Descripción de las cuotas de memoria

Parámetro	Descripción
Memory request	Cantidad mínima de memoria requerida por un contenedor. Los recursos se programan para el contenedor en función de este valor. El contenedor se puede programar para este nodo solamente cuando la memoria total disponible en el nodo es mayor o igual que el número de aplicaciones de memoria en contenedores.
Memory Limit	Cantidad máxima de memoria disponible para un contenedor. Cuando el uso de memoria excede el límite de memoria configurado, la instancia puede reiniciarse, lo que afecta al uso normal de la carga de trabajo.

### Configuración recomendada

Memoria real disponible de un nodo  $\geq$  Suma de los límites de memoria de todos los contenedores del nodo actual  $\geq$  Suma de las peticiones de memoria de todos los contenedores del nodo actual. Puede ver la memoria disponible real de un nodo en la consola de CCE (**Resource Management > Nodes > Allocatable**).

#### NOTA

Los recursos asignables se calculan basándose en el valor **Request** de solicitud de recurso, que indica el límite superior de recursos que pueden solicitarse por los pods en este nodo, pero no indica los recursos disponibles reales del nodo (para más detalles, véase **Ejemplo**). La fórmula de cálculo es la siguiente:

- CPU asignable = CPU total - CPU solicitada de todos los pods - CPU reservada para otros recursos
- Memoria asignable = Memoria total - Memoria solicitada de todos los pods - Memoria reservada para otros recursos

## Ejemplo

Suponga que un clúster contiene un nodo con 4 núcleos de CPU y 8 GiB de memoria. Se han desplegado dos pods (pod 1 y pod 2) en el clúster. Pod 1 sobresuscribe recursos (es decir, **Limit > Request**). Las especificaciones de los dos pods son las siguientes:

Pod	Solicitud de CPU	Límite de CPU	Solicitud de memoria	Límite de memoria
Pod 1	1 núcleo	2 núcleos	2 GiB	4 GiB
Pod 2	2 núcleos	2 núcleos	2 GiB	2 GiB

El uso de CPU y memoria del nodo es el siguiente:

- CPUs asignables = 4 núcleos (1 núcleo solicitado por el pod 1 + 2 núcleos solicitados por el pod 2) = 1 núcleo
- Memoria asignable = 8 GiB – (1 GiB solicitado por el pod 1 + 2 GiB solicitado por el pod 2) = 5 GiB

En este caso, el resto de 1 núcleo de 5 GiB puede ser utilizado por el siguiente nuevo pod.

Si el pod 1 está bajo carga pesada durante las horas pico, usará más CPU y memoria dentro del límite. Por lo tanto, los recursos asignables reales son menores que 1 núcleo de 5 GiB.

## 5.3.4 Setting Container Lifecycle Parameters

### Scenario

CCE provides callback functions for the lifecycle management of containerized applications. For example, if you want a container to perform a certain operation before stopping, you can register a hook function.

CCE provides the following lifecycle callback functions:

- **Startup Command:** executed to start a container. For details, see [Startup Commands](#).
- **Post-Start:** executed immediately after a container is started. For details, see [Post-Start Processing](#).
- **Pre-Stop:** executed before a container is stopped. The pre-stop processing function helps you ensure that the services running on the pods can be completed in advance in the case of pod upgrade or deletion. For details, see [Pre-Stop Processing](#).

### Startup Commands

By default, the default command during image start. To run a specific command or rewrite the default image value, you must perform specific settings:

A Docker image has metadata that stores image information. If lifecycle commands and arguments are not set, CCE runs the default commands and arguments, that is, Docker instructions **ENTRYPOINT** and **CMD**, provided during image creation.

If the commands and arguments used to run a container are set during application creation, the default commands **ENTRYPOINT** and **CMD** are overwritten during image build. The rules are as follows:

**Tabla 5-9** Commands and arguments used to run a container

Image ENTRYPOINT	Image CMD	Command to Run a Container	Parameters to Run a Container	Command Executed
[touch]	[/root/test]	Not set	Not set	[touch /root/test]
[touch]	[/root/test]	[mkdir]	Not set	[mkdir]
[touch]	[/root/test]	Not set	[/opt/test]	[touch /opt/test]
[touch]	[/root/test]	[mkdir]	[/opt/test]	[mkdir /opt/test]

**Paso 1** Log in to the CCE console. When creating a workload, configure container information and select **Lifecycle**.

**Paso 2** Enter a command and arguments on the **Startup Command** tab page.

**Tabla 5-10** Container startup command

Configuration Item	Procedure
Command	Enter an executable command, for example, <b>/run/server</b> . If there are multiple commands, separate them with spaces. If the command contains a space, you need to add a quotation mark (""). <b>NOTA</b> In the case of multiple commands, you are advised to run <b>/bin/sh</b> or other <b>shell</b> commands. Other commands are used as parameters.
Args	Enter the argument that controls the container running command, for example, <b>--port=8080</b> . If there are multiple arguments, separate them in different lines.

---Fin

## Post-Start Processing

**Paso 1** Log in to the CCE console. When creating a workload, configure container information and select **Lifecycle**.

**Paso 2** Set the post-start processing parameters on the **Post-Start** tab page.

**Tabla 5-11** Post-start processing parameters

Parameter	Description
CLI	<p>Set commands to be executed in the container for post-start processing. The command format is <b>Command Args[1] Args[2]...</b>. <b>Command</b> is a system command or a user-defined executable program. If no path is specified, an executable program in the default path will be selected. If multiple commands need to be executed, write the commands into a script for execution. <b>Commands that are executed in the background or asynchronously are not supported.</b></p> <p>Example command:</p> <pre>exec:   command:     - /install.sh     - install_agent</pre> <p>Enter <b>/install install_agent</b> in the script. This command indicates that <b>install.sh</b> will be executed after the container is created successfully.</p>
HTTP request	<p>Send an HTTP request for post-start processing. The related parameters are described as follows:</p> <ul style="list-style-type: none"> <li>● <b>Path:</b> (optional) request URL.</li> <li>● <b>Port:</b> (mandatory) request port.</li> <li>● <b>Host:</b> (optional) IP address of the request. The default value is the IP address of the node where the container resides.</li> </ul>

---Fin

## Pre-Stop Processing

- Paso 1** Log in to the CCE console. When creating a workload, configure container information and select **Lifecycle**.
- Paso 2** Set the pre-start processing parameters on the **Pre-Stop** tab page.

**Tabla 5-12** Pre-stop processing parameters

Parameter	Description
CLI	<p>Set commands to be executed in the container for pre-stop processing. The command format is <b>Command Args[1] Args[2]...</b> <b>Command</b> is a system command or a user-defined executable program. If no path is specified, an executable program in the default path will be selected. If multiple commands need to be executed, write the commands into a script for execution.</p> <p>Example command:</p> <pre>exec:   command:   - /uninstall.sh   - uninstall_agent</pre> <p>Enter <b>/uninstall uninstall_agent</b> in the script. This command indicates that the <b>uninstall.sh</b> script will be executed before the container completes its execution and stops running.</p>
HTTP request	<p>Send an HTTP request for pre-stop processing. The related parameters are described as follows:</p> <ul style="list-style-type: none"> <li>● <b>Path:</b> (optional) request URL.</li> <li>● <b>Port:</b> (mandatory) request port.</li> <li>● <b>Host:</b> (optional) IP address of the request. The default value is the IP address of the node where the container resides.</li> </ul>

---Fin

## Example YAML

This section uses Nginx as an example to describe how to set the container lifecycle.

In the following configuration file, the **postStart** command is defined to run the **install.sh** command in the **/bin/bash** directory. **preStop** is defined to run the **uninstall.sh** command.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx
        command:
        - sleep 3600 #Startup command
        imagePullPolicy: Always
        lifecycle:
          postStart:
            exec:
              command:
```

```
- /bin/bash
- install.sh #Post-start command
preStop:
  exec:
    command:
      - /bin/bash
      - uninstall.sh #Pre-stop command
name: nginx
imagePullSecrets:
- name: default-secret
```

## 5.3.5 Configuración de la comprobación de estado de un contenedor

### Escenario

Comprobación de estado comprueba regularmente el estado de salud de contenedores durante la ejecución de contenedor. Si la función de comprobación de estado no está configurada, un pod no puede detectar excepciones de aplicación ni reiniciar automáticamente la aplicación para restaurarla. Esto dará como resultado una situación en la que el estado del pod es normal pero la aplicación en el pod es anormal.

Kubernetes proporciona las siguientes sondas de comprobación de estado:

- **Liveness probe** (livenessProbe): comprueba si un contenedor sigue vivo. Es similar al comando **ps** que comprueba si existe un proceso. Si se produce un error en la comprobación de vida de un contenedor, el clúster reinicia el contenedor. Si la comprobación de vitalidad tiene éxito, no se ejecuta ninguna operación.
- **Readiness probe** (readinessProbe): comprueba si un contenedor está listo para procesar las solicitudes de los usuarios. Al detectarse que el contenedor no está listo, el tráfico de servicio no se dirigirá al contenedor. Algunas aplicaciones pueden tardar mucho tiempo en iniciarse antes de que puedan proporcionar servicios. Esto se debe a que necesitan cargar datos de disco o confiar en el inicio de un módulo externo. En este caso, el proceso de aplicación se está ejecutando, pero la aplicación no puede proporcionar servicios. Para solucionar este problema, se utiliza este sondeo de comprobación de estado. Si la comprobación de disponibilidad del contenedor falla, el clúster enmascara todas las solicitudes enviadas al contenedor. Si la comprobación de la preparación del contenedor se realiza correctamente, se puede acceder al contenedor.
- **Startup probe** (startupProbe): comprueba cuando se ha iniciado una aplicación de contenedor. Si se configura un sondeo de este tipo, deshabilita las comprobaciones de disponibilidad y de vida hasta que tenga éxito, asegurando que esos sondeos no interfieran con el inicio de la aplicación. Esto se puede usar para adoptar controles de vida en contenedores de inicio lento, evitando que el kubelet los termine antes de que se inicien.

### Método de comprobación

- **HTTP request**

Este modo de comprobación de estado es aplicable a contenedores que proporcionan servicios HTTP/HTTPS. El clúster inicia periódicamente una solicitud GET de HTTP/HTTPS a tales contenedores. Si el código de retorno de la respuesta HTTP/HTTPS está dentro de 200-399, la sonda tiene éxito. De lo contrario, la sonda falla. En este modo de comprobación de estado, debe especificar un puerto de escucha contenedor y una ruta de solicitud HTTP/HTTPS.

Por ejemplo, para un contenedor que proporciona servicios HTTP, la ruta de comprobación HTTP es **/health-check** y el puerto es 80 y la dirección del host es opcional (que por defecto es la dirección IP de contenedor). Aquí, 172.16.0.186 se usa como ejemplo, y podemos obtener tal petición: GET http://172.16.0.186:80/health-check. El clúster inicia periódicamente esta solicitud al contenedor. También puede agregar uno o más encabezados a una solicitud de HTTP. Por ejemplo, establezca el nombre del encabezado de la solicitud en **Custom-Header** y el valor correspondiente en **example**.

**Figura 5-2** Comprobación basada en solicitudes HTTP

The screenshot shows the 'Liveness Probe' configuration page. On the left, there is a section for 'Enable' (checked), 'Check Method' (HTTP selected), 'Path' (/health-check), 'Port' (80), 'Host Address' (172.16.0.186), and 'Protocol' (HTTP selected). On the right, there is a section for 'Period (s)' (10), 'Delay (s)' (0), 'Timeout (s)' (1), 'Success Threshold' (1), 'Failure Threshold' (3), and 'Request Header' (empty).

● **TCP port**

Para un contenedor que proporciona servicios de comunicación de TCP, el clúster establece periódicamente una conexión TCP con contenedor. Si la conexión se realiza correctamente, la sonda se realiza correctamente. De lo contrario, la sonda falla. En este modo de comprobación de estado, debe especificar un puerto de escucha de contenedor. Por ejemplo, si tiene un contenedor Nginx con el puerto de servicio 80, después de especificar el puerto TCP 80 para la escucha de contenedor, el clúster iniciará periódicamente una conexión TCP con el puerto 80 del contenedor. Si la conexión se realiza correctamente, la sonda se realiza correctamente. De lo contrario, la sonda falla.

**Figura 5-3** Comprobación basada en el puerto de TCP

The screenshot shows the 'Liveness Probe' configuration page. On the left, there is a section for 'Enable' (checked), 'Check Method' (TCP selected), and 'Port' (80). On the right, there is a section for 'Period (s)' (10), 'Delay (s)' (0), 'Timeout (s)' (1), 'Success Threshold' (1), and 'Failure Threshold' (3).

● **CLI**

CLI es una herramienta eficiente para la comprobación de estado. Cuando utilice CLI, debe especificar un comando ejecutable en un contenedor. El clúster ejecuta periódicamente el comando en contenedor. Si el resultado del comando es 0, la comprobación de estado se realiza correctamente. De lo contrario, la comprobación de estado falla.

El modo de CLI se puede utilizar para reemplazar la comprobación de estado basada en petición HTTP y basada en el puerto de TCP.

- Para un puerto de TCP, puede utilizar un script de programa para conectarse a un puerto de contenedor. Si la conexión se realiza correctamente, el script devuelve **0**. De lo contrario, la secuencia de comandos devuelve **-1**.

- Para una solicitud de HTTP, puede usar el comando `script` para ejecutar el comando `wget` para detectar el contenedor.

**wget http://127.0.0.1:80/health-check**

Compruebe el código de retorno de la respuesta. Si el código de retorno está dentro de 200-399, la secuencia de comandos devuelve **0**. De lo contrario, la secuencia de comandos devuelve **-1**.

**Figura 5-4** Comprobación basada en CLI

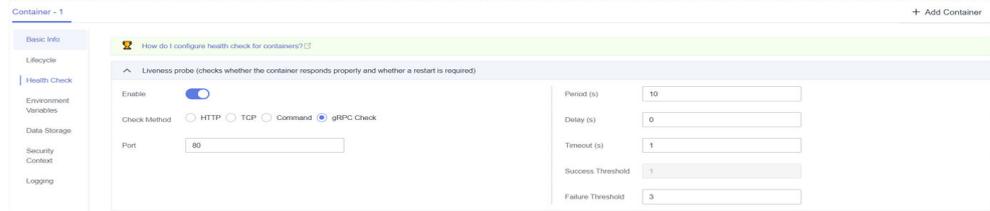
### AVISO

- Ponga el programa a ejecutar en la imagen contenedor para que el programa pueda ser ejecutado.
  - Si el comando que se va a ejecutar es un script de shell, no especifique directamente el script como el comando, sino agregue un analizador de scripts. Por ejemplo, si el script es `/data/scripts/health_check.sh` debe especificar `sh/data/scripts/health_check.sh` para la ejecución del comando. La razón es que el clúster no está en el entorno de terminal cuando se ejecutan programas en un contenedor.
- 
- **gRPC Check**  
Las comprobaciones de gRPC pueden configurar sondas de inicio, vida y preparación para su aplicación gRPC sin exponer ningún punto de conexión de HTTP, ni necesita un ejecutable. Kubernetes puede conectarse a su carga de trabajo a través de gRPC y consultar su estado.

### AVISO

- La comprobación de gRPC solo se admite en los clústeres de CCE de v1.25 o posterior.
- Para usar gRPC para verificación, su aplicación debe soportar el **protocolo de verificación de estado de gRPC**.
- Similar a los sondeos HTTP y TCP, si el puerto es incorrecto o la aplicación no admite el protocolo de comprobación de estado, la comprobación falla.

**Figura 5-5 Comprobación de gRPC**



## Parámetros comunes

**Tabla 5-13** Descripción del parámetro común

Parámetro	Descripción
<b>Period</b> (periodSeconds)	Indica el período de detección de la sonda, en segundos. Por ejemplo, si este parámetro se establece en <b>30</b> , la detección se realiza cada 30 segundos.
<b>Delay</b> (initialDelaySeconds)	Compruebe el tiempo de retraso en segundos. Establezca este parámetro de acuerdo con la hora normal de inicio de los servicios. Por ejemplo, si este parámetro se establece en <b>30</b> , la comprobación de estado se iniciará 30 segundos después de iniciar el contenedor. El tiempo está reservado para que comiencen los servicios en contenedores.
<b>Timeout</b> (timeoutSeconds)	Número de segundos después de los cuales la sonda se agota. Unidad: segundo. Por ejemplo, si este parámetro se establece en <b>10</b> , el tiempo de espera para realizar una comprobación de estado es de 10s. Si el tiempo de espera transcurre, la comprobación de estado se considera una falla. Si el parámetro se deja en blanco o se establece en <b>0</b> , el tiempo de espera predeterminado es de 1s.
<b>Success Threshold</b> (successThreshold)	Mínimo éxitos consecutivos para que la sonda se considere exitosa después de haber fallado. Por ejemplo, si este parámetro se establece en <b>1</b> , el estado de la carga de trabajo es normal solo cuando la comprobación de estado es correcta durante una vez consecutiva después de que la comprobación de estado falla. El valor predeterminado es <b>1</b> , que también es el valor mínimo. El valor de este parámetro se fija a <b>1</b> en las <b>Liveness Probe</b> y <b>Startup Probe</b> .
<b>Failure Threshold</b> (failureThreshold)	Número de reintentos cuando falla la detección. Renunciarse en caso de sonda de vida significa reiniciar el contenedor. En caso de sonda de preparación, el pod estará marcada como Unready. El valor predeterminado es <b>3</b> . El valor mínimo es <b>1</b> .

## Ejemplo de YAML

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    test: liveness
    name: liveness-http
spec:
  containers:
  - name: liveness
    image: nginx:alpine
    args:
    - /server
    livenessProbe:
      httpGet:
        path: /healthz
        port: 80
        httpHeaders:
        - name: Custom-Header
          value: Awesome
      initialDelaySeconds: 3
      periodSeconds: 3
    readinessProbe:
      exec:
        command:
        - cat
        - /tmp/healthy
      initialDelaySeconds: 5
      periodSeconds: 5
    startupProbe:
      httpGet:
        path: /healthz
        port: 80
      failureThreshold: 30
      periodSeconds: 10
```

### 5.3.6 Setting an Environment Variable

#### Scenario

An environment variable is a variable whose value can affect the way a running container will behave. You can modify environment variables even after workloads are deployed, increasing flexibility in workload configuration.

The function of setting environment variables on CCE is the same as that of specifying **ENV** in a Dockerfile.

---

#### AVISO

After a container is started, do not modify configurations in the container. If configurations in the container are modified (for example, passwords, certificates, and environment variables of a containerized application are added to the container), the configurations will be lost after the container restarts and container services will become abnormal. An example scenario of container restart is pod rescheduling due to node anomalies.

Configurations must be imported to a container as arguments. Otherwise, configurations will be lost after the container restarts.

---

Environment variables can be set in the following modes:

- **Custom**
- **Added from ConfigMap:** Import all keys in a ConfigMap as environment variables.
- **Added from ConfigMap key:** Import a key in a ConfigMap as the value of an environment variable. For example, if you import **configmap\_value** of **configmap\_key** in a ConfigMap as the value of environment variable **key1**, an environment variable named **key1** with its value **is configmap\_value** exists in the container.
- **Added from secret:** Import all keys in a secret as environment variables.
- **Added from secret key:** Import the value of a key in a secret as the value of an environment variable. For example, if you import **secret\_value** of **secret\_key** in secret **secret-example** as the value of environment variable **key2**, an environment variable named **key2** with its value **secret\_value** exists in the container.
- **Variable value/reference:** Use the field defined by a pod as the value of the environment variable, for example, the pod name.
- **Resource Reference:** Use the field defined by a container as the value of the environment variable, for example, the CPU limit of the container.

## Adding Environment Variables

**Paso 1** Log in to the CCE console. When creating a workload, select **Environment Variables** under **Container Settings**.

**Paso 2** Set environment variables.

Type	Variable Name	Variable Value/Reference	Operation
Custom	key	value	Delete
Added from ConfigMap key	key1	configmap-example configmap_key	Delete
Added from secret key	key2	secret-example secret-key	Delete
Variable Value/Reference	key3	metadata.name	Delete
Resource Reference	key4	container-1 limits.cpu	Delete
Added from ConfigMap		configmap-example	Delete
Added from secret		secret-example	Delete

----Fin

## YAML Example

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: env-example
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      app: env-example
  template:
    metadata:
      labels:
        app: env-example
    spec:
      containers:
        - name: container-1
          image: nginx:alpine
          imagePullPolicy: Always
          resources:
    
```

```

requests:
  cpu: 250m
  memory: 512Mi
limits:
  cpu: 250m
  memory: 512Mi
env:
  - name: key                               # Custom
    value: value
  - name: key1                              # Added from ConfigMap key
    valueFrom:
      configMapKeyRef:
        name: configmap-example
        key: key1
  - name: key2                              # Added from secret key
    valueFrom:
      secretKeyRef:
        name: secret-example
        key: key2
  - name: key3                              # Variable reference, which uses the
field defined by a pod as the value of the environment variable.
    valueFrom:
      fieldRef:
        apiVersion: v1
        fieldPath: metadata.name
  - name: key4                              # Resource reference, which uses the
field defined by a container as the value of the environment variable.
    valueFrom:
      resourceFieldRef:
        containerName: container1
        resource: limits.cpu
        divisor: 1
envFrom:
  - configMapRef:                           # Added from ConfigMap
    name: configmap-example
  - secretRef:                              # Added from secret
    name: secret-example
imagePullSecrets:
  - name: default-secret
    
```

## Viewing Environment Variables

If the contents of **configmap-example** and **secret-example** are as follows:

```

$ kubectl get configmap configmap-example -oyaml
apiVersion: v1
data:
  configmap_key: configmap_value
kind: ConfigMap
...

$ kubectl get secret secret-example -oyaml
apiVersion: v1
data:
  secret_key: c2VjcmV0X3ZhbHV1             # c2VjcmV0X3ZhbHV1 is the value of
secret_value in Base64 mode.
kind: Secret
...
    
```

The environment variables in the pod are as follows:

```

$ kubectl get pod
NAME                                READY   STATUS    RESTARTS   AGE
env-example-695b759569-1x9jp        1/1     Running   0           17m

$ kubectl exec env-example-695b759569-1x9jp -- printenv
/ # env
key=value                             # Custom environment variable
eyl=configmap_value                    # Added from ConfigMap key
    
```

```
key2=secret_value # Added from secret key
key3=env-example-695b759569-1x9jp # metadata.name defined by the pod
key4=1 # limits.cpu defined by container1. The
value is rounded up, in unit of cores.
configmap_key=configmap_value # Added from ConfigMap. The key value in
the original ConfigMap key is directly imported.
secret_key=secret_value # Added from key. The key value in the
original secret is directly imported.
```

## 5.3.7 Configuración de la configuración de APM para el análisis de cuello de botella del rendimiento

### Escenario

Application Performance Management (APM) le permite supervisar las cargas de trabajo de Java mediante el seguimiento y la topología. Puede instalar sondas APM para localizar y analizar problemas de cargas de trabajo de Java.

Puede configurar la supervisión de cargas de trabajo de Java cuando y después de crear una carga de trabajo.

#### NOTA

- Necesita conectar su aplicación Java en CCE a Application Performance Management (APM) con la sonda Pinpoint. Para obtener más información, consulte [Conexión de una aplicación en contenedores en Huawei Cloud a APM](#).

### Requisitos previos

Si no ha habilitado el servicio de APM, vaya a la consola de APM y habilítela como se le indique.

### Procedimiento

**Paso 1** Inicie sesión en la consola de CCE.

**Paso 2** En la ficha **APM Settings** del área **Advanced Settings**, el sondeo está deshabilitado de forma predeterminada. Puede seleccionar **APM1.0 probe** necesario. Después de activar la sonda, APM puede localizar y analizar los problemas de los programas Java.

#### NOTA

1. La sonda APM 1.0 se inicializará en un contenedor de inicio creado automáticamente llamado init-pinpoint. Al contenedor de inicio se le asignará una CPU de 0.25 núcleos y una memoria de 250 MiB.
2. Al agregar un sondeo APM se agregarán las variables de entorno PAAS\_MONITORING\_GROUP, JAVA\_TOOL\_OPTIONS y PAAS\_CLUSTER\_ID a todos los contenedores de servicio.
3. Al agregar una sonda APM, se montará un volumen de almacenamiento local llamado paas-apm (para sonda APM 1.0) en todos los contenedores de servicio.

**Paso 3** Establezca los parámetros relacionados con la sonda.

#### Sonda APM1.0

- **Monitoring Group:** Introduzca un nombre de grupo de supervisión, por ejemplo, **testapp**. Si hay uno o más grupos de supervisión disponibles, puede seleccionar uno de la lista desplegable.

- **Probe Version:** Seleccione la versión de la sonda.
- **Probe Upgrade Policy:** De forma predeterminada, se selecciona **Auto upgrade upon restart**.
  - **Auto upgrade upon restart:** El sistema descarga la imagen de la sonda cada vez que se reinicia el pod.
  - **Manual upgrade upon restart:** Esta política significa que si hay una imagen local disponible, se usará la imagen local. El sistema descarga la imagen de sondeo solo cuando una imagen local no está disponible.

**Paso 4** Tres minutos después de iniciar la aplicación, sus datos se mostrarán en la consola de APM. Puede iniciar sesión en la consola de APM y optimizar el rendimiento de las aplicaciones mediante topología y seguimiento. Para obtener más información, véase [Topología](#).

---Fin

## 5.3.8 Habilitación de reglas de grupo de seguridad ICMP

### Escenario

Si una carga de trabajo utiliza UDP tanto para el balanceo de carga como para la comprobación de estado, habilite las reglas de grupo de seguridad ICMP para los servidores backend.

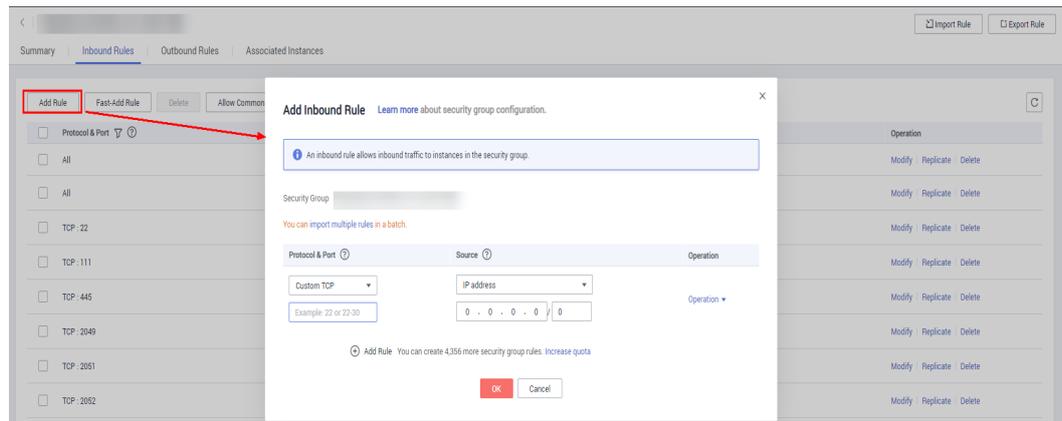
### Procedimiento

- Paso 1** Inicie sesión en la consola de ECS, busque el ECS correspondiente a cualquier nodo donde se ejecute la carga de trabajo y haga clic en el nombre de ECS. En la página de detalles de ECS mostrada, registre el nombre del grupo de seguridad.
- Paso 2** Inicie sesión en la consola de VPC. En el panel de navegación de la izquierda, elija **Access Control > Security Groups**. En la lista de grupos de seguridad de la derecha, haga clic en el nombre del grupo de seguridad obtenido en el paso 1.
- Paso 3** En la página mostrada, haga clic en la ficha **Inbound Rules** y haga clic en **Add Rule** para agregar una regla de entrada para ECS. A continuación, haga clic en **OK**. Para obtener más información sobre cómo agregar una regla entrante para ECS, vea [Figura 5-6](#).

#### NOTA

- Solo necesita agregar reglas de grupo de seguridad a cualquier nodo en el que se ejecute la carga de trabajo.
- El grupo de seguridad debe tener reglas para permitir el acceso desde el bloque CIDR 100.125.0.0/16.

**Figura 5-6** Incorporación de una regla de grupo de seguridad



----Fin

### 5.3.9 Configuración de una política de extracción de imágenes

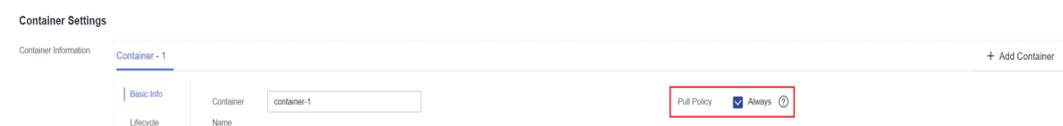
Cuando se crea una carga de trabajo, la imagen de contenedor se extrae del repositorio de imágenes al nodo. La imagen también se extrae cuando se reinicia o se actualiza la carga de trabajo.

De forma predeterminada, se establece **imagePullPolicy** en **IfNotPresent** e indica que si la imagen existe en el nodo, se utiliza la imagen existente. Si la imagen no existe en el nodo, la imagen se extrae del repositorio de imágenes.

La política de extracción de imagen también se puede establecer en **Always**, lo que indica que la imagen se extrae del repositorio de imágenes y sobrescribe la imagen en el nodo independientemente de si la imagen existe en el nodo.

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - image: nginx:alpine
    name: container-0
    resources:
      limits:
        cpu: 100m
        memory: 200Mi
      requests:
        cpu: 100m
        memory: 200Mi
    imagePullPolicy: Always
  imagePullSecrets:
  - name: default-secret
```

También puede establecer la política de extracción de imágenes al crear una carga de trabajo en la consola de CCE. Como se muestra en la siguiente figura, si selecciona **Always**, siempre se tira de la imagen. Si no lo selecciona, la política será **IfNotPresent**, lo que significa que la imagen no se extrae.



**AVISO**

Se recomienda utilizar una nueva etiqueta cada vez que cree una imagen. Si no actualiza la etiqueta, pero solo actualiza la imagen, cuando **Pull Policy** se establece en **IfNotPresent**, CCE considera que ya existe una imagen con la etiqueta en el nodo actual y no volverá a extraer la imagen.

### 5.3.10 Configuración de la sincronización de zona horaria

Al crear una carga de trabajo, puede configurar contenedores para que utilice la misma zona horaria que el nodo. Puede habilitar la sincronización de zona horaria al crear una carga de trabajo.

Time Zone Synchronization



If this setting is enabled, the same time zone will be used for both containers and nodes.

La función de sincronización de zona horaria depende del disco local (hostPath) montado en el contenedor. Después de habilitar la sincronización de zona horaria, **/etc/localtime** del nodo se monta a **/etc/localtime** del contenedor en modo HostPath. De esta manera, el nodo y el contenedor utilizan el mismo archivo de configuración de zona horaria.

```
kind: Deployment
apiVersion: apps/v1
metadata:
  name: test
  namespace: default
spec:
  replicas: 2
  selector:
    matchLabels:
      app: test
  template:
    metadata:
      labels:
        app: test
    spec:
      volumes:
        - name: vol-162979628557461404
          hostPath:
            path: /etc/localtime
            type: ''
      containers:
        - name: container-0
          image: 'nginx:alpine'
          volumeMounts:
            - name: vol-162979628557461404
              readOnly: true
              mountPath: /etc/localtime
              imagePullPolicy: IfNotPresent
          imagePullSecrets:
            - name: default-secret
```

### 5.3.11 Configuración de la política de actualización de carga de trabajo

En aplicaciones reales, la actualización es una operación común. Deployment, StatefulSet o DaemonSet pueden admitir fácilmente la actualización de la aplicación.

Puede establecer diferentes políticas de actualización:

- **Rolling upgrade:** Los pods nuevos se crean gradualmente y luego los pods antiguos se eliminan. Esta es la política predeterminada.
- **Replace upgrade:** Los pods actuales se eliminan y luego se crean nuevos pods.

The screenshot shows a configuration interface for upgrading workloads. It features two tabs: 'Rolling upgrade' (selected) and 'Replace upgrade'. Below the tabs, there are several input fields and their descriptions:

- Upgrade Mode:** A note states 'Old pods will be gradually replaced with new pods. During the upgrade, services will be evenly distributed to both old and new pods to ensure service continuity.'
- Max. Unavailable:** A text input field containing '25' and a dropdown menu set to '%'. Description: 'Maximum number of unavailable pods allowed in a rolling upgrade. If the number is equal to the total number of pods, services may be interrupted. (Minimum number of alive pods = Total pods - Maximum number of unavailable pods)'
- Max. Surge:** A text input field containing '25' and a dropdown menu set to '%'. Description: 'Maximum number of pods that can be created over the desired number of pods in each rolling upgrade'
- Min. Ready Seconds:** A text input field containing '0'. Description: 'Minimum number of seconds for which a newly created pod should be ready without any of its containers crashing, for it to be considered available.'
- Revision History Limit:** A text input field containing '10'. Description: 'Limit'
- Max. Upgrade Duration (s):** A text input field containing '600'. Description: 'Duration (s)'
- Scale-In Time:** A text input field containing '30'. Description: 'Scale-In Time'
- Window (s):** A text input field containing '30'. Description: 'Time window (0-5999s) for pre-stop commands to finish execution before a workload is forcibly deleted. value is 30s.'

## Parámetros de actualización

- **Max. Surge (maxSurge)**  
Especifica el número máximo de pods que **spec.replicas** puede existir. El valor predeterminado es 25%. Por ejemplo, si **spec.replicas** se establece en **4**, no pueden existir más de 5 pods durante el proceso de actualización, es decir, el paso de actualización es 1. El número absoluto se calcula a partir del porcentaje redondeando al alza. El valor también se puede establecer en un número absoluto.  
Este parámetro solo es compatible con Deployments.
- **Max. Unavailable Pods (maxUnavailable)**  
Especifica el número máximo de pods que pueden no estar disponibles durante el proceso de actualización. El valor predeterminado es 25%. Por ejemplo, si **spec.replicas** se establece en **4** existen al menos 3 pods durante el proceso de actualización, es decir, el paso de eliminación es 1. El valor también se puede establecer en un número absoluto.  
Este parámetro solo es compatible con Deployments.
- **Min. Ready Seconds (minReadySeconds)**  
Un pod se considera disponible solo cuando se excede el tiempo mínimo de preparación sin que se bloquee ninguno de sus contenedores. El valor predeterminado es **0** (el pod se considera disponible inmediatamente después de que está lista).
- **Revision History Limit (revisionHistoryLimit)**  
Especifica el número de ReplicaSets antiguas que se conservarán para permitir la reversión. Estas ReplicaSets antiguas consumen recursos en etcd y llenan la salida de **kubectl get rs**. La configuración de cada revisión de Deployment se almacena en su ReplicaSets. Por lo tanto, una vez que se elimina la ReplicaSet antigua, se pierde la capacidad de volver a esa revisión de la Deployment. Por defecto, se mantendrán 10 ReplicaSets antiguas, pero el valor ideal depende de la frecuencia y estabilidad de las nuevas implementaciones.
- **Max. Upgrade Duration (progressDeadlineSeconds)**  
Especifica el número de segundos que el sistema espera a que una Deployment progrese antes de informar de un error de progreso de Deployment. Se presenta como una condición con **Type=Progressing, Status=False y Reason=ProgressDeadlineExceeded** en el estado del recurso. El controlador de Deployment seguirá reintentando la Deployment. En el futuro, una vez que se implemente la reversión automática, el controlador de Deployment revertirá una Deployment tan pronto como observe dicha condición.  
Si se especifica este parámetro, el valor de este parámetro debe ser mayor que el de **.spec.minReadySeconds**.

- **Scale-In Time Window** (`terminationGracePeriodSeconds`)

Tiempo de eliminación elegante. El valor predeterminado es de 30 segundos. Cuando se elimina un pod, se envía una señal SIGTERM y el sistema espera a que terminen las aplicaciones en el contenedor. Si la aplicación no termina dentro del tiempo especificado por `terminationGracePeriodSeconds` se envía una señal SIGKILL para terminar por la fuerza el pod.

## Ejemplo de actualización

La Deployment se puede actualizar en un modo declarativo. Es decir, solo necesita modificar la definición de YAML de la Deployment. Por ejemplo, puede ejecutar el comando `kubectl edit` para cambiar la imagen de Deployment a `nginx:alpine`. Después de la modificación, consulte el ReplicaSet y el pod. El resultado de la consulta muestra que se crea un nuevo ReplicaSet y se vuelve a crear el pod.

```
$ kubectl edit deploy nginx

$ kubectl get rs
NAME                DESIRED   CURRENT   READY   AGE
nginx-6f9f58dfffd   2         2         2       1m
nginx-7f98958cdf    0         0         0       48m

$ kubectl get pods
NAME                READY     STATUS    RESTARTS   AGE
nginx-6f9f58dfffd-tdmqk  1/1      Running   0          1m
nginx-6f9f58dfffd-tesqr  1/1      Running   0          1m
```

La Deployment puede usar los parámetros `maxSurge` y `maxUnavailable` para controlar la proporción de pods que se van a recrear durante la actualización, lo que es útil en muchos escenarios. La configuración es la siguiente:

```
spec:
  strategy:
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 0
    type: RollingUpdate
```

En el ejemplo anterior, el valor de `spec.replicas` es de 2. Si `maxSurge` y `maxUnavailable` son el valor predeterminado 25%, `maxSurge` permite que existan un máximo de tres pods ( $2 \times 1.25 = 2.5$ , redondeado a 3), y `maxUnavailable` no permite que no estén disponibles un máximo de dos pods ( $2 \times 0.75 = 1.5$ , redondeado a 2). Es decir, durante el proceso de actualización, siempre habrá dos pods en ejecución. Cada vez que se crea un nuevo pod, se elimina un pod antiguo hasta que todos los pods sean nuevos.

## Retroceso

El retroceso consiste en revertir una aplicación a la versión anterior cuando se produce un error durante la actualización. Una Deployment se puede volver fácilmente a la versión anterior.

Por ejemplo, si la imagen actualizada es defectuosa, puede ejecutar el comando `kubectl rollout undo` para revertir la Deployment.

```
$ kubectl rollout undo deployment nginx
deployment.apps/nginx rolled back
```

Una Deployment se puede revertir fácilmente porque utiliza un ReplicaSet para controlar un pod. Después de la actualización, el ReplicaSet anterior todavía existe. La Deployment se revierte mediante el ReplicaSet anterior para volver a crear el pod. El parámetro

**revisionHistoryLimit** puede restringir el número de ReplicaSets almacenados en una Deployment. El valor predeterminado es **10**.

### 5.3.12 Política de programación (afinidad/antiafinidad)

Un nodeSelector proporciona una forma muy sencilla de restringir pods a nodos con etiquetas particulares, como se menciona en [Creación de un DaemonSet](#). La función de afinidad y antiafinidad amplía enormemente los tipos de restricciones que puede expresar.

Kubernetes admite afinidad y antiafinidad a nivel de nodo y de pod. Puede configurar reglas personalizadas para lograr una programación de afinidad y antiafinidad. For example, you can deploy frontend pods and backend pods together, deploy the same type of applications on a specific node, or deploy different applications on different nodes.

#### Node Affinity (nodeAffinity)

Labels are the basis of affinity rules. Let's look at the labels on nodes in a cluster.

```
$ kubectl describe node 192.168.0.212
Name:          192.168.0.212
Roles:         <none>
Labels:        beta.kubernetes.io/arch=amd64
               beta.kubernetes.io/os=linux
               failure-domain.beta.kubernetes.io/is-baremetal=false
               failure-domain.beta.kubernetes.io/region=*****
               failure-domain.beta.kubernetes.io/zone=*****
               kubernetes.io/arch=amd64
               kubernetes.io/availablezone=*****
               kubernetes.io/eniquota=12
               kubernetes.io/hostname=192.168.0.212
               kubernetes.io/os=linux
               node.kubernetes.io/subnetid=fd43acad-33e7-48b2-
a85a-24833f362e0e
               os.architecture=amd64
               os.name=EulerOS_2.0_SP5
               os.version=3.10.0-862.14.1.5.h328.eulerosv2r7.x86_64
```

Estas etiquetas son agregadas automáticamente por CCE durante la creación del nodo. A continuación se describen algunas que se utilizan con frecuencia durante la programación.

- **failure-domain.beta.kubernetes.io/region**: región donde se encuentra el nodo.
- **failure-domain.beta.kubernetes.io/zone**: zona de disponibilidad a la que pertenece el nodo.
- **kubernetes.io/hostname**: nombre de host del nodo.

Cuando despliega pods, puede usar un nodeSelector, como se describe en el documento [DaemonSet](#) para restringir pods a nodos con las etiquetas específicas. En el ejemplo siguiente se muestra cómo utilizar un nodeSelector para desplegar pods solo en los nodos con la etiqueta **gpu=true**.

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  nodeSelector:
    # Node selection. A pod is deployed on a node
    # only when the node has the gpu=true label.
    gpu: true
  ...
```

Las reglas de afinidad de nodo pueden lograr los mismos resultados, como se muestra en el siguiente ejemplo.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: gpu
  labels:
    app: gpu
spec:
  selector:
    matchLabels:
      app: gpu
  replicas: 3
  template:
    metadata:
      labels:
        app: gpu
    spec:
      containers:
      - image: nginx:alpine
        name: gpu
        resources:
          requests:
            cpu: 100m
            memory: 200Mi
          limits:
            cpu: 100m
            memory: 200Mi
      imagePullSecrets:
      - name: default-secret
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
            - matchExpressions:
              - key: gpu
                operator: In
                values:
                - "true"
```

Aunque la regla de afinidad de nodo requiere más líneas, es más expresiva, lo que se describirá más adelante.

**requiredDuringSchedulingIgnoredDuringExecution** parece ser complejo, pero puede entenderse fácilmente como una combinación de dos partes.

- **requiredDuringScheduling** indica que los pods se pueden programar en el nodo solo cuando se cumplen todas las reglas definidas (requerido).
- **IgnoredDuringExecution** indica que los pods que ya se están ejecutando en el nodo no necesitan cumplir las reglas definidas. Es decir, se ignora una etiqueta en el nodo, y los pods que requieren que el nodo contenga esa etiqueta no se reprogramarán.

Además, el valor de **operator** es **In**, indicando que el valor de la etiqueta debe estar en la lista de valores. Otros valores de operador disponibles son los siguientes:

- **NotIn**: El valor de etiqueta no está en una lista.
- **Exists**: Existe una etiqueta específica.
- **DoesNotExist**: No existe una etiqueta específica.
- **Gt**: El valor de etiqueta es mayor que un valor especificado (comparación de cadenas).
- **Lt**: El valor de etiqueta es menor que un valor especificado (comparación de cadenas).

Tenga en cuenta que no hay tal cosa como **nodeAntiAffinity** porque los operadores **NotIn** y **DoesNotExist** proporcionan la misma función.

A continuación se describe cómo comprobar si la regla tiene efecto. Suponga que un clúster tiene tres nodos.

```
$ kubectl get node
NAME                STATUS    ROLES    AGE   VERSION
192.168.0.212      Ready    <none>   13m   v1.15.6-r1-20.3.0.2.B001-15.30.2
192.168.0.94       Ready    <none>   13m   v1.15.6-r1-20.3.0.2.B001-15.30.2
192.168.0.97       Ready    <none>   13m   v1.15.6-r1-20.3.0.2.B001-15.30.2
```

Agregue la etiqueta **gpu=true** al nodo **192.168.0.212**.

```
$ kubectl label node 192.168.0.212 gpu=true
node/192.168.0.212 labeled

$ kubectl get node -L gpu
NAME                STATUS    ROLES    AGE   VERSION    GPU
192.168.0.212      Ready    <none>   13m   v1.15.6-r1-20.3.0.2.B001-15.30.2    true
192.168.0.94       Ready    <none>   13m   v1.15.6-r1-20.3.0.2.B001-15.30.2
192.168.0.97       Ready    <none>   13m   v1.15.6-r1-20.3.0.2.B001-15.30.2
```

Cree la Deployment. Puede encontrar que todos los pods se despliegan en el nodo **192.168.0.212**.

```
$ kubectl create -f affinity.yaml
deployment.apps/gpu created

$ kubectl get pod -o wide
NAME                READY    STATUS    RESTARTS   AGE   IP
NODE
gpu-6df65c44cf-42xw4    1/1     Running    0           15s   172.16.0.37
192.168.0.212
gpu-6df65c44cf-jzjvs    1/1     Running    0           15s   172.16.0.36
192.168.0.212
gpu-6df65c44cf-zv5c1    1/1     Running    0           15s   172.16.0.38
192.168.0.212
```

## Regla de preferencia de nodo

La regla **requiredDuringSchedulingIgnoredDuringExecution** anterior es una regla de selección dura. Hay otro tipo de regla de selección, es decir, **preferredDuringSchedulingIgnoredDuringExecution**. Se utiliza para especificar qué nodos se prefieren durante la planificación.

Para lograr este efecto, agregue un nodo conectado con discos de SAS al clúster, agregue la etiqueta **DISK=SAS** al nodo y agregue la etiqueta **DISK=SSD** a los otros tres nodos.

```
$ kubectl get node -L DISK,gpu
NAME                STATUS    ROLES    AGE   VERSION    DISK    GPU
192.168.0.100      Ready    <none>   7h23m   v1.15.6-r1-20.3.0.2.B001-15.30.2    SAS
192.168.0.212      Ready    <none>   8h      v1.15.6-r1-20.3.0.2.B001-15.30.2    SSD    true
192.168.0.94       Ready    <none>   8h      v1.15.6-r1-20.3.0.2.B001-15.30.2
192.168.0.97       Ready    <none>   8h      v1.15.6-r1-20.3.0.2.B001-15.30.2    SSD
```

Defina una Deployment. Utilice la regla **preferredDuringSchedulingIgnoredDuringExecution** para establecer la ponderación de los nodos con el disco de SSD instalado como **80** y los nodos con la etiqueta **gpu=true** como **20**. De esta manera, los pods se despliegan preferentemente en los nodos con el disco SSD instalado.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: gpu
  labels:
    app: gpu
```

```
spec:
  selector:
    matchLabels:
      app: gpu
  replicas: 10
  template:
    metadata:
      labels:
        app: gpu
    spec:
      containers:
      - image: nginx:alpine
        name: gpu
        resources:
          requests:
            cpu: 100m
            memory: 200Mi
          limits:
            cpu: 100m
            memory: 200Mi
      imagePullSecrets:
      - name: default-secret
      affinity:
        nodeAffinity:
          preferredDuringSchedulingIgnoredDuringExecution:
          - weight: 80
            preference:
              matchExpressions:
              - key: DISK
                operator: In
                values:
                - SSD
          - weight: 20
            preference:
              matchExpressions:
              - key: gpu
                operator: In
                values:
                - "true"
```

Después del despliegue, hay cinco pods desplegados en el nodo **192.168.0.212** (etiqueta: **DISK=SSD** y **GPU=true**), tres pods desplegados en el nodo **192.168.0.97** (etiqueta: **DISK=SSD**), y dos pods desplegados en el nodo **192.168.0.100** (etiqueta: **DISK=SAS**).

En la salida anterior, puede encontrar que ningún pod de la Deployment está programado al nodo **192.168.0.94** (etiqueta: **DISK=SSD**). Esto se debe a que el nodo ya tiene muchos pods en él y su uso de recursos es alto. Esto también indica que la regla **preferredDuringSchedulingIgnoredDuringExecution** define una preferencia en lugar de un requisito difícil.

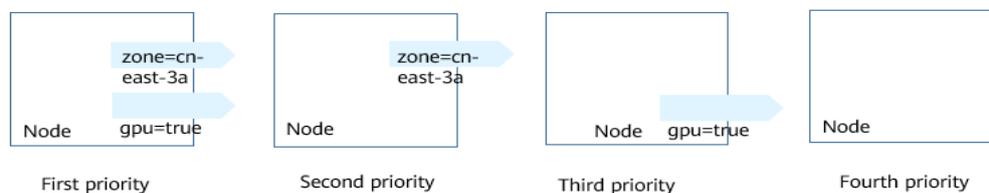
```
$ kubectl create -f affinity2.yaml
deployment.apps/gpu created

$ kubectl get po -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP
NODE
gpu-585455d466-5bmcz                1/1     Running  0           2m29s 172.16.0.44
192.168.0.212
gpu-585455d466-cg2l6                1/1     Running  0           2m29s 172.16.0.63
192.168.0.97
gpu-585455d466-f2bt2                1/1     Running  0           2m29s 172.16.0.79
192.168.0.100
gpu-585455d466-hdb5n                1/1     Running  0           2m29s 172.16.0.42
192.168.0.212
gpu-585455d466-hkgvz                1/1     Running  0           2m29s 172.16.0.43
192.168.0.212
gpu-585455d466-mngvn                1/1     Running  0           2m29s 172.16.0.48
192.168.0.97
```

gpu-585455d466-s26qs 192.168.0.97	1/1	Running	0	2m29s	172.16.0.62
gpu-585455d466-sxtzm 192.168.0.212	1/1	Running	0	2m29s	172.16.0.45
gpu-585455d466-t56cm 192.168.0.100	1/1	Running	0	2m29s	172.16.0.64
gpu-585455d466-t5w5x 192.168.0.212	1/1	Running	0	2m29s	172.16.0.41

En el ejemplo anterior, la prioridad de planificación de nodo es como sigue. Los nodos con etiquetas **SSD** y **gpu=true** tienen la prioridad más alta. Los nodos con la etiqueta **SSD** pero sin etiqueta **gpu=true** tienen la segunda prioridad (peso: 80). Los nodos con la etiqueta **gpu=true** pero sin etiqueta **SSD** tienen la tercera prioridad. Los nodos sin ninguna de estas dos etiquetas tienen la prioridad más baja.

Figura 5-7 Planificación de prioridades



## Afinidad de la carga de trabajo (podAffinity)

Las reglas de afinidad de nodos afectan solo a la afinidad entre pods y nodos. Kubernetes también admite la configuración de reglas de afinidad dentro de pod. Por ejemplo, el frontend y el backend de una aplicación se pueden desplegar juntos en un nodo para reducir la latencia de acceso. También hay dos tipos de reglas de afinidad dentro de pod:

**requiredDuringSchedulingIgnoredDuringExecution** y **preferredDuringSchedulingIgnoredDuringExecution**.

Suponga que el backend de una aplicación ha sido creado y tiene la etiqueta **app=backend**.

```
$ kubectl get po -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP
NODE
backend-658f6cb858-dlrz8           1/1    Running   0          2m36s  172.16.0.67
192.168.0.100
```

Puede configurar la siguiente regla de afinidad de pod para desplegar los pods frontend de la aplicación en el mismo nodo que los pods backend.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: frontend
  labels:
    app: frontend
spec:
  selector:
    matchLabels:
      app: frontend
  replicas: 3
  template:
    metadata:
      labels:
        app: frontend
    spec:
      containers:
        - image: nginx:alpine
```

```

name: frontend
resources:
  requests:
    cpu: 100m
    memory: 200Mi
  limits:
    cpu: 100m
    memory: 200Mi
imagePullSecrets:
- name: default-secret
affinity:
  podAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
    - topologyKey: kubernetes.io/hostname
      labelSelector:
        matchExpressions:
        - key: app
          operator: In
          values:
          - backend
    
```

Despliegue el frontend y puede encontrar que el frontend se despliega en el mismo nodo que el backend.

```

$ kubectl create -f affinity3.yaml
deployment.apps/frontend created

$ kubectl get po -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP
NODE
backend-658f6cb858-dlrz8            1/1     Running   0           5m38s 172.16.0.67
192.168.0.100
frontend-67ff9b7b97-dsqzn          1/1     Running   0           6s     172.16.0.70
192.168.0.100
frontend-67ff9b7b97-hxm5t          1/1     Running   0           6s     172.16.0.71
192.168.0.100
frontend-67ff9b7b97-z8pdb          1/1     Running   0           6s     172.16.0.72
192.168.0.100
    
```

El campo **topologyKey** se utiliza para dividir dominios de topología para especificar el rango de selección. Si las claves de etiqueta y los valores de los nodos son los mismos, se considera que los nodos están en el mismo dominio de topología. A continuación, se seleccionan los contenidos definidos en las siguientes reglas. El efecto de **topologyKey** no se demuestra completamente en el ejemplo anterior porque todos los nodos tienen la etiqueta **kubernetes.io/hostname**. Es decir, todos los nodos están dentro del rango.

Para ver cómo funciona **topologyKey**, suponga que el backend de la aplicación tiene dos pods, que se ejecutan en los nodos diferentes.

```

$ kubectl get po -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP
NODE
backend-658f6cb858-5bpd6            1/1     Running   0           23m   172.16.0.40
192.168.0.97
backend-658f6cb858-dlrz8            1/1     Running   0           2m36s 172.16.0.67
192.168.0.100
    
```

Agregue la etiqueta **prefer=true** a los nodos **192.168.0.97** y **192.168.0.94**.

```

$ kubectl label node 192.168.0.97 prefer=true
node/192.168.0.97 labeled
$ kubectl label node 192.168.0.94 prefer=true
node/192.168.0.94 labeled

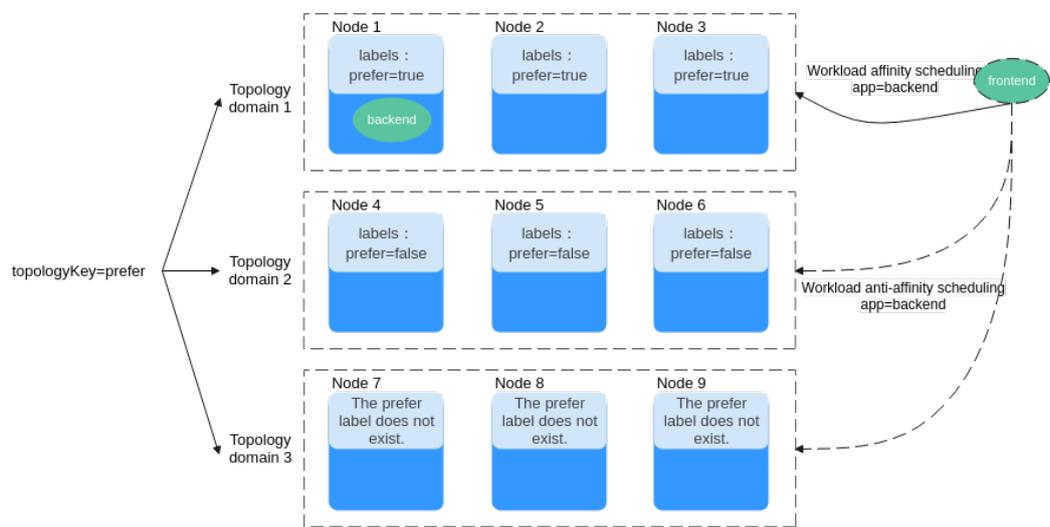
$ kubectl get node -L prefer
NAME                                STATUS    ROLES    AGE   VERSION                                PREFER
192.168.0.100                       Ready    <none>   44m   v1.15.6-r1-20.3.0.2.B001-15.30.2
192.168.0.212                       Ready    <none>   91m   v1.15.6-r1-20.3.0.2.B001-15.30.2
    
```

192.168.0.94	Ready	<none>	91m	v1.15.6-r1-20.3.0.2.B001-15.30.2	true
192.168.0.97	Ready	<none>	91m	v1.15.6-r1-20.3.0.2.B001-15.30.2	true

Si la **topologyKey** de **podAffinity** se establece en **prefer**, los dominios de topología de nodo se dividen como se muestra en **Figura 5-8**.

```
affinity:
  podAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
    - topologyKey: prefer
      labelSelector:
        matchExpressions:
        - key: app
          operator: In
          values:
          - backend
```

**Figura 5-8** Dominios de topología



Durante la planificación, los dominios de topología de nodo se dividen en función de la etiqueta **prefer**. En este ejemplo, las **192.168.0.97** y **192.168.0.94** se dividen en el mismo dominio de topología. Si un pod con la etiqueta **app=backend** se ejecuta en el dominio de topología, aunque no todos los nodos del dominio de topología ejecuten el pod con la etiqueta **app=backend** (en este ejemplo, solo el nodo **192.168.0.97** tiene tal pod) **frontend** también se despliega en este dominio de topología (**192.168.0.97** o **192.168.0.94**).

```
$ kubectl create -f affinity3.yaml
deployment.apps/frontend created

$ kubectl get po -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP
NODE
backend-658f6cb858-5bpd6            1/1    Running   0          26m   172.16.0.40
192.168.0.97
backend-658f6cb858-dlrz8            1/1    Running   0          5m38s 172.16.0.67
192.168.0.100
frontend-67ff9b7b97-dsqzn          1/1    Running   0          6s    172.16.0.70
192.168.0.97
frontend-67ff9b7b97-hxm5t          1/1    Running   0          6s    172.16.0.71
192.168.0.97
frontend-67ff9b7b97-z8pdb          1/1    Running   0          6s    172.16.0.72
192.168.0.97
```

## Antiafinidad de carga de trabajo (podAntiAffinity)

A diferencia de los escenarios en los que se prefiere que los pods se planifiquen en el mismo nodo, a veces, podría ser exactamente lo contrario. Por ejemplo, si ciertos pods se despliegan juntos, afectarán al rendimiento.

El siguiente es un ejemplo de definición de una regla antiafinidad. Esta regla divide los dominios de topología de nodo por la etiqueta **kubernetes.io/hostname**. Si ya existe un pod con la etiqueta **app=frontend** en un nodo del dominio de topología, los pods con la misma etiqueta no se pueden programar en otros nodos del dominio de topología.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: frontend
  labels:
    app: frontend
spec:
  selector:
    matchLabels:
      app: frontend
  replicas: 5
  template:
    metadata:
      labels:
        app: frontend
    spec:
      containers:
      - image: nginx:alpine
        name: frontend
        resources:
          requests:
            cpu: 100m
            memory: 200Mi
          limits:
            cpu: 100m
            memory: 200Mi
      imagePullSecrets:
      - name: default-secret
      affinity:
        podAntiAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
          - topologyKey: kubernetes.io/hostname # Topology domain of the node
            labelSelector: # Pod label matching rule
              matchExpressions:
              - key: app
                operator: In
                values:
                - frontend
```

Cree una regla de antiafinidad y vea el resultado despliegue. En el ejemplo, los dominios de topología de nodo se dividen por la etiqueta **kubernetes.io/hostname**. Los valores de etiqueta de los nodos con la etiqueta **kubernetes.io/hostname** son diferentes, por lo que solo hay un nodo en un dominio de topología. Si ya existe un pod con la etiqueta **frontend** en un dominio de topología (un nodo en este ejemplo), el dominio de topología no programará pods con la misma etiqueta. En este ejemplo, solo hay cuatro nodos. Por lo tanto, hay un pod que está en el estado **Pending** y no se puede programar.

```
$ kubectl create -f affinity4.yaml
deployment.apps/frontend created

$ kubectl get po -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP
NODE
frontend-6f686d8d87-8dlsc          1/1     Running   0           18s   172.16.0.76
192.168.0.100
```

frontend-6f686d8d87-d618p	0/1	Pending	0	18s	<none>
<none>					
frontend-6f686d8d87-hgcq2	1/1	Running	0	18s	172.16.0.54
192.168.0.97					
frontend-6f686d8d87-q7cfq	1/1	Running	0	18s	172.16.0.47
192.168.0.212					
frontend-6f686d8d87-x18hx	1/1	Running	0	18s	172.16.0.23
192.168.0.94					

## Configuración de políticas de programación

**Paso 1** Inicie sesión en la consola de CCE.

**Paso 2** Cuando cree una carga de trabajo, haga clic en **Scheduling** en el área **Advanced Settings**.

**Tabla 5-14** Configuración de afinidad de nodo

Parámetro	Descripción
Required	Esta es una regla dura que se debe cumplir para la programación. Corresponde a <b>requiredDuringSchedulingIgnoredDuringExecution</b> en Kubernetes. Se pueden establecer varias reglas requeridas, y la programación se realizará si solo se cumple una de ellas.
Preferred	Esta es una regla flexible que especifica las preferencias que el planificador intentará aplicar, pero no garantiza. Corresponde a <b>preferredDuringSchedulingIgnoredDuringExecution</b> en Kubernetes. La programación se realiza cuando se cumple una regla o cuando no se cumple ninguna de las reglas.

**Paso 3** En **Node Affinity**, **Workload Affinity** y **Workload Anti-Affinity**, haga clic en **+** para agregar políticas de programación. En el cuadro de diálogo que se muestra, agregue una política directamente o especificando un nodo o una AZ.

La especificación de un nodo o una AZ se implementa esencialmente con etiquetas. La etiqueta **kubernetes.io/hostname** se utiliza cuando se especifica un nodo y la etiqueta **failure-domain.beta.kubernetes.io/zone** se utiliza cuando se especifica una AZ.

**Tabla 5-15** Configuración de políticas de programación

Parámetro	Descripción
Label	Etiqueta del nodo. Puede utilizar la etiqueta predeterminada o personalizar una etiqueta.

Parámetro	Descripción
Operator	Se apoyan las siguientes relaciones: <b>In</b> , <b>NotIn</b> , <b>Exists</b> , <b>DoesNotExist</b> , <b>Gt</b> y <b>Lt</b> <ul style="list-style-type: none"> <li>● <b>In</b>: Existe una etiqueta en la lista de etiquetas.</li> <li>● <b>NotIn</b>: Una etiqueta no existe en la lista de etiquetas.</li> <li>● <b>Exists</b>: Existe una etiqueta específica.</li> <li>● <b>DoesNotExist</b>: No existe una etiqueta específica.</li> <li>● <b>Gt</b>: El valor de etiqueta es mayor que un valor especificado (comparación de cadenas).</li> <li>● <b>Lt</b>: El valor de etiqueta es menor que un valor especificado (comparación de cadenas).</li> </ul>
Label Value	Valor de etiqueta.
Namespace	Este parámetro solo está disponible en una política de programación de afinidad o antiafinidad de carga de trabajo.  Espacio de nombres para el que entra en vigor la política de programación.
Topology Key	Este parámetro solo se puede utilizar en una política de programación de afinidad o antiafinidad de carga de trabajo.  Seleccione el ámbito especificado por <b>topologyKey</b> y, a continuación, seleccione el contenido definido por la política.
Weight	Este parámetro solo se puede establecer en una política de programación de <b>Preferred</b> .

----Fin

## 5.3.13 Pod Scale-in Priorities

### Scale-in Priorities

Pod scale-in is performed based on the following priorities:

1. Pods that are not scheduled
2. Pod Pending < Pod Unknown < Pod Running
3. Not ready < Ready
4. Pod with a specific label (cce.io/priordelation)
5. Doubled up < Not doubled up (nodes with more pods are first downsized)
6. Pods that take a long time to be ready for scale-in
7. Pods that are frequently restarted
8. Pods with an earlier **createtime**

### Preferential Scale-in of Old Pods

- Where such scale-in **does not work**:

According to the preceding priorities, assume that the number of nodes is 2 (the same specifications) and the number of pods in a Deployment is 5 (or any odd number), three pods distributed on node A and two on node B. If the number of pods is reduced to 3, **Rule 5** is performed first, and the system determines that the node to be scaled in is node A. Then **Rule 6** is performed. After an old pod is deleted, it is found that no old pod exists and the scale-in node remains node A. In this case, a new pod will be deleted.

- **Where such scale-in works:** Assume that the number of nodes is 2 (the same specifications) and the number of Deployment pods is 6 (or any even number). These pods are distributed evenly on nodes A and B. If the number of pods is reduced to 4, the system determines that the **Rule 5** is not met. In this case, both nodes A and B will be scaled in. Then, the system performs **Rule 6** to sort the pods on the two nodes and deletes two old pods.

### 5.3.14 Etiquetas y anotaciones de pod

#### Anotaciones de Pod

CCE le permite agregar anotaciones a un archivo YAML para realizar algunas funciones avanzadas de pod. En la siguiente tabla se describen las anotaciones que se pueden agregar.

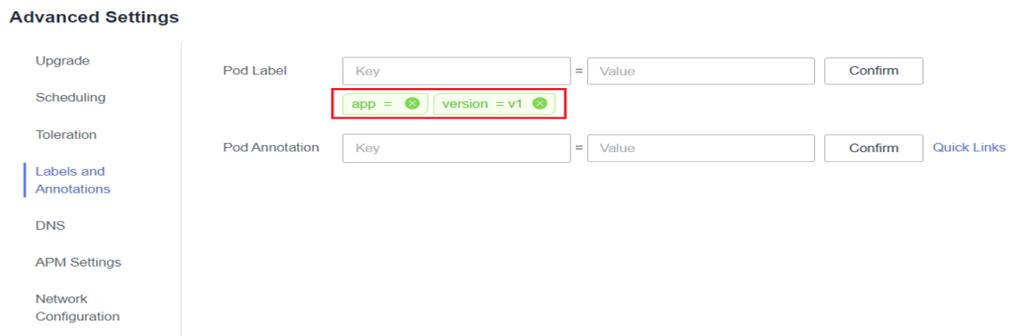
**Tabla 5-16** Anotaciones de pod

Anotación	Descripción	Valor predeterminado
kubernetes.AOM.log.stdout	<p>Parámetro de salida estándar. Si no se especifica, la salida de log estándar de todas los contenedores se notifica a la AOM. Puede recopilar logs de stdout de los contenedores ciertos o ignorarlos en absoluto.</p> <p>Por ejemplo:</p> <ul style="list-style-type: none"> <li>● Recopilación de ninguno de los logs de stdout: kubernetes.AOM.log.stdout: '[]'</li> <li>● Recopilación de logs de stdout de contenedor-1 y contenedor-2: kubernetes.AOM.log.stdout: '["container-1","container-2"]'</li> </ul>	-
metrics.alpha.kubernetes.io/custom-endpoints	<p>Parámetro para informar de las métricas de supervisión de AOM que especifique.</p> <p>Para obtener más información, véase <a href="#">Supervisión de métricas personalizadas en AOM</a>.</p>	-

Anotación	Descripción	Valor predeterminado
prometheus.io/scrape	Parámetro para informar métricas de Prometheus. Si el valor es de <b>true</b> , la carga de trabajo actual informa de las métricas de supervisión. Para obtener más información, véase <a href="#">Monitoreo de métricas personalizadas con prometheus</a> .	-
prometheus.io/path	URL para que Prometheus recopile datos. Para obtener más información, véase <a href="#">Monitoreo de métricas personalizadas con prometheus</a> .	/metrics
prometheus.io/port	Número de puerto de punto de conexión para que Prometheus recopile datos. Para obtener más información, véase <a href="#">Monitoreo de métricas personalizadas con prometheus</a> .	-
prometheus.io/scheme	Protocolo utilizado por Prometheus para recopilar datos. El valor puede ser <b>http</b> o <b>https</b> . Para obtener más información, véase <a href="#">Monitoreo de métricas personalizadas con prometheus</a> .	-
kubernetes.io/ingress-bandwidth	Ancho de banda de entrada de un pod. Para obtener más información, véase <a href="#">Configuración de la limitación de la velocidad de QoS para el acceso entre los pod</a> .	-
kubernetes.io/egress-bandwidth	Ancho de banda de salida de un pod. Para obtener más información, véase <a href="#">Configuración de la limitación de la velocidad de QoS para el acceso entre los pod</a> .	-

## Etiquetas de pod

Al crear una carga de trabajo en la consola, se agregan las siguientes etiquetas al pod de forma predeterminada. El valor de **app** es el nombre de la carga de trabajo.



Ejemplo de YAML:

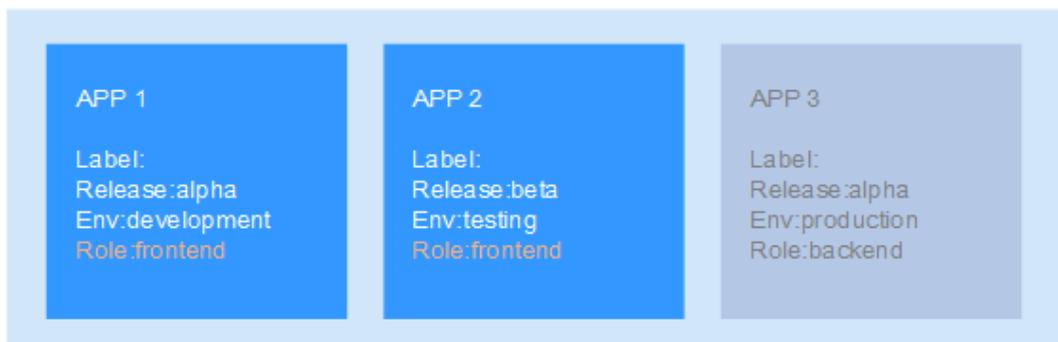
```

...
spec:
  selector:
    matchLabels:
      app: nginx
      version: v1
  template:
    metadata:
      labels:
        app: nginx
        version: v1
    spec:
      ...
    
```

También puede agregar otras etiquetas al pod para la programación de afinidad y antiafinidad. En la siguiente figura, se definen tres etiquetas de pod (release, env y role) para la carga de trabajo APP 1, APP 2 y APP 3. Los valores de estas etiquetas varían según la carga de trabajo.

- APP 1: [release:alpha;env:development;role:frontend]
- APP 2: [release:beta;env:testing;role:frontend]
- APP 3: [release:alpha;env:production;role:backend]

Figura 5-9 Ejemplo de etiqueta



Por ejemplo, si el valor de **key/value** es **role/backend**, se seleccionará la APP 3 para la planificación de afinidad. Para obtener más información, véase [Afinidad de la carga de trabajo \(podAffinity\)](#).

## 5.4 Gestión de cargas de trabajo y trabajos

### Escenario

Después de crear una carga de trabajo, puede actualizar, supervisar, revertir o eliminar la carga de trabajo, así como editar su archivo YAML.

**Tabla 5-17** Gestión de cargas de trabajo/trabajo

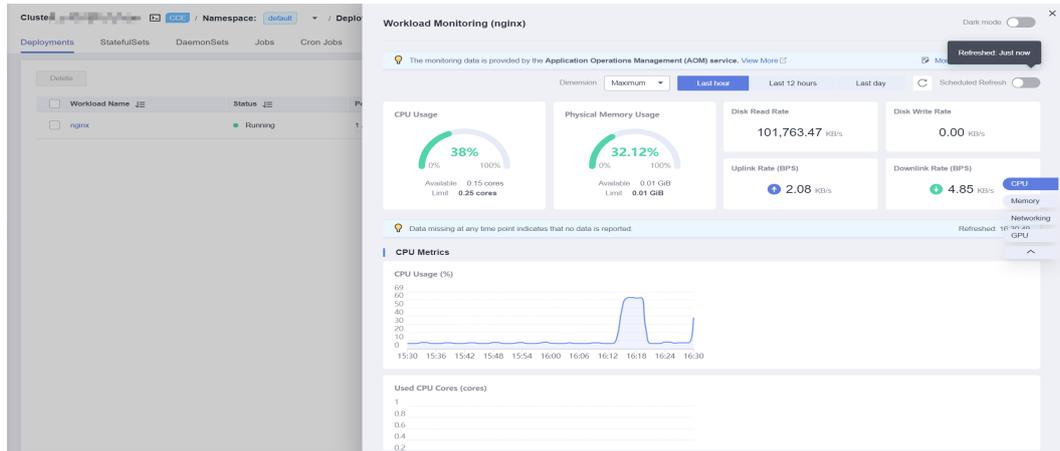
Operación	Descripción
<b>Monitoreo</b>	Puede ver el uso de CPU y memoria de las cargas de trabajo y los pods en la consola de CCE.
<b>Ver log</b>	Puede ver los logs de las cargas de trabajo.
<b>Actualización</b>	Puede reemplazar imágenes o etiquetas de imagen para actualizar rápidamente Deployments, StatefulSets y DaemonSets sin interrumpir los servicios.
<b>Editar YAML</b>	Puede modificar y descargar los archivos YAML de StatefulSets, DaemonSets y los pods en la consola de CCE. Los archivos YAML de trabajos y trabajos cron solo se pueden ver, copiar y descargar.
<b>Retroceder</b>	Solo se pueden revertir las Deployments.
<b>Redespliegue</b>	Puede volver a desplegar una carga de trabajo. Después de que se redespliegue la carga de trabajo, se reiniciarán todos los pods de la carga de trabajo.
<b>Activación/ desactivación de la actualización</b>	Solo Deployments admiten esta operación.
<b>Gestionar etiqueta</b>	Las etiquetas se adjuntan a las cargas de trabajo como pares de clave-valor para gestionar y seleccionar cargas de trabajo. Jobs y Cron Jobs no admiten esta operación.
<b>Eliminar</b>	Puede eliminar una carga de trabajo o un trabajo que ya no sea necesario. Las cargas de trabajo o trabajos eliminados no se pueden recuperar.
<b>Ver eventos</b>	Puede ver los nombres de eventos, los tipos de eventos, el número de ocurrencias, los eventos de Kubernetes, la hora de primera ocurrencia y la hora de última ocurrencia.
<b>Detener/Iniciar</b>	Solo puede iniciar o detener un trabajo cron.

### Supervisión de una carga de trabajo

Puede ver el uso de CPU y memoria de Deployments y pods en la consola de CCE para determinar las especificaciones de recursos que puede necesitar. En esta sección se utiliza una Deployment como ejemplo para describir cómo supervisar una carga de trabajo.

- Paso 1** Inicie sesión en la consola de CCE, vaya a un clúster existente y elija **Workloads** en el panel de navegación.
- Paso 2** Haga clic en la ficha **Deployments** y haga clic en **Monitor** de la carga de trabajo de destino. En la página que se muestra, puede ver el uso de la CPU y el uso de memoria de la carga de trabajo.

**Figura 5-10** Consulta de información de supervisión



- Paso 3** Haga clic en el nombre de la carga de trabajo. En la página de ficha **Pods**, haga clic en el **Monitor** del pod de destino para ver su uso de CPU y memoria.

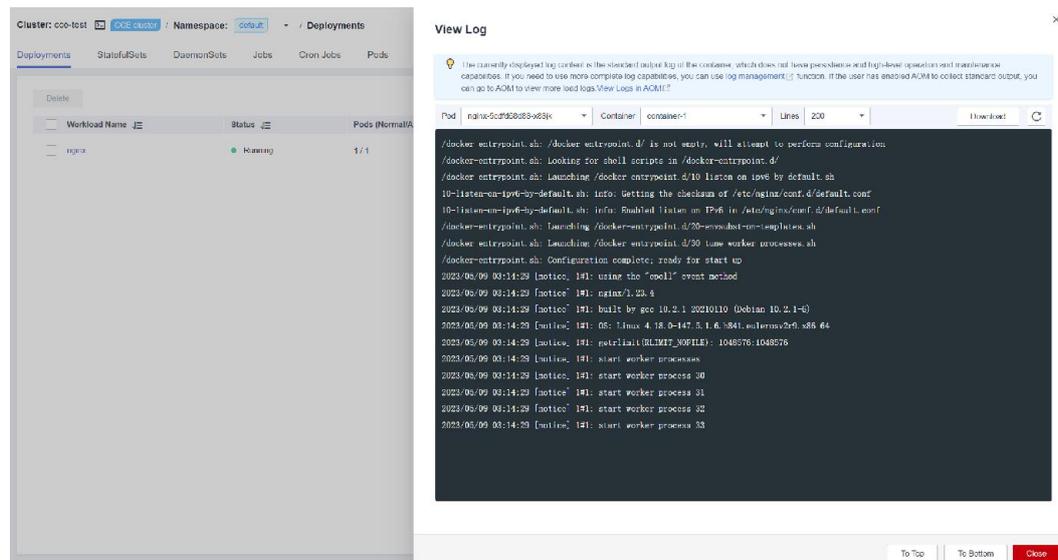
----Fin

## Visualización de logs

Puede ver los logs de Deployments, de StatefulSets, de DaemonSets y de los trabajos. En esta sección se utiliza una Deployment como ejemplo para describir cómo ver los logs.

- Paso 1** Inicie sesión en la consola de CCE, vaya a un clúster existente y elija **Workloads** en el panel de navegación.
- Paso 2** Haga clic en la ficha **Deployments** y haga clic en el **View Log** de la carga de trabajo de destino.  
En la ventana **View Log** mostrada, puede ver los logs.

**Figura 5-11** Consulta de logs de una carga de trabajo



**NOTA**

Los logs mostrados son los de salida estándar de contenedores y no tienen persistencia y capacidades avanzadas de O&M. Para utilizar capacidades de log más completas, consulte [Logs](#). Si la función de recopilación de salida estándar está habilitada para la carga de trabajo (habilitada de forma predeterminada), puede ir a AOM para ver más logs de carga de trabajo. Para obtener más información, véase [Uso de ICAgent para recopilar logs de contenedores](#).

----Fin

## Actualización de una carga de trabajo

Puede actualizar rápidamente Deployments, StatefulSets y DaemonSets en la consola de CCE.

En esta sección se utiliza una Deployment como ejemplo para describir cómo actualizar una carga de trabajo.

Antes de reemplazar una imagen o una versión de imagen, cargue la nueva imagen en el servicio SWR. Para obtener más información, consulte [Carga de una imagen a través de un cliente de motor de contenedores](#).

- Paso 1** Inicie sesión en la consola de CCE, vaya a un clúster existente y elija **Workloads** en el panel de navegación.
- Paso 2** Haga clic en la ficha **Deployments** y haga clic en **Upgrade** de la carga de trabajo de destino.

**NOTA**

- Las cargas de trabajo no se pueden actualizar por lotes.
- Antes de realizar una actualización de StatefulSet in situ, debe eliminar manualmente los pods antiguos. De lo contrario, el estado de actualización siempre se muestra como **Upgrading**.

- Paso 3** Actualice la carga de trabajo en función de los requisitos de servicio. El método para establecer el parámetro es el mismo que para crear una carga de trabajo.

**Paso 4** Una vez completada la actualización, haga clic en **Upgrade Workload**, confirme manualmente el archivo YAML y envíe la actualización.

----Fin

## Edición de un archivo YAML

Puede modificar y descargar los archivos YAML de StatefulSets, DaemonSets y los pods en la consola de CCE. Los archivos YAML de trabajos y trabajos cron solo se pueden ver, copiar y descargar. Esta sección utiliza una Deployment como ejemplo para describir cómo editar el archivo YAML.

**Paso 1** Inicie sesión en la consola de CCE, vaya a un clúster existente y elija **Workloads** en el panel de navegación.

**Paso 2** Haga clic en la ficha **Deployments** y elija **More > Edit YAML** en la columna **Operation** de la carga de trabajo de destino. En el cuadro de diálogo que se muestra, modifique el archivo YAML.

**Paso 3** Haga clic en **Edit** y luego en **OK** para guardar los cambios.

**Paso 4** (Opcional) En la ventana **Edit YAML**, haga clic en **Download** para descargar el archivo YAML.

----Fin

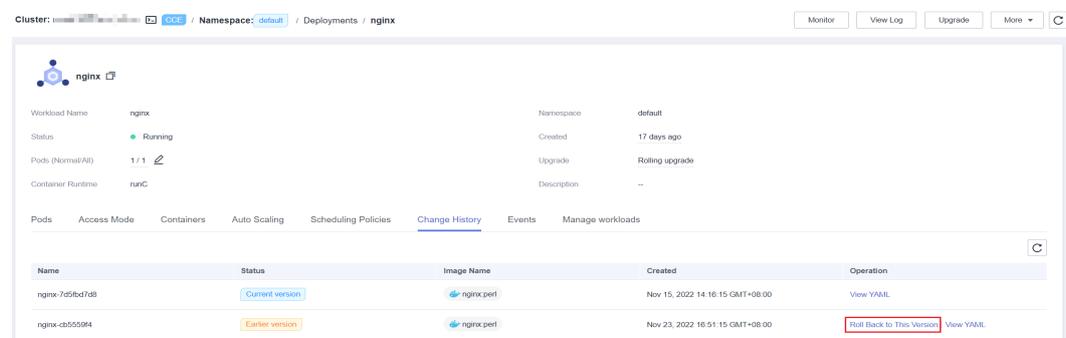
## Revertir una carga de trabajo (disponible solo para Deployments)

CCE registra el historial de versiones de todas las Deployments. Puede revertir una Deployment a una versión especificada.

**Paso 1** Inicie sesión en la consola de CCE, vaya a un clúster existente y elija **Workloads** en el panel de navegación.

**Paso 2** Haga clic en la ficha **Deployments** y elija **More > Roll Back** en la columna **Operation** de la carga de trabajo de destino.

**Paso 3** Cambie a la página de ficha **Change History**, haga clic en **Roll Back to This Version** de la versión de destino, confirme manualmente el archivo YAML y haga clic en **OK**.



----Fin

## Redesplegar una carga de trabajo

Después de volver a desplegar una carga de trabajo, se reiniciarán todos los pods de la carga de trabajo. En esta sección se utilizan Deployments como ejemplo para ilustrar cómo volver a desplegar una carga de trabajo.

- Paso 1** Inicie sesión en la consola de CCE, vaya a un clúster existente y elija **Workloads** en el panel de navegación.
- Paso 2** Haga clic en la ficha **Deployments** y elija **More > Redeploy** en la columna **Operation** de la carga de trabajo de destino.
- Paso 3** En el cuadro de diálogo que se muestra, haga clic en **Yes** para volver a desplegar la carga de trabajo.

----Fin

## Deshabilitar/habilitar la actualización (disponible solo para Deployments)

Solo las Deployments admiten esta operación.

- Después de deshabilitar la actualización, el comando upgrade se puede entregar, pero no se aplicará a los pods.  
Si está realizando una actualización sucesiva, la actualización sucesiva se detiene después de que se entregue el comando de desactivación de actualización. En este caso, los nuevos y viejos pods coexisten.
- Si se está actualizando una Deployment, se puede actualizar o revertir. Sus pods heredarán las últimas actualizaciones de la Deployment. Si son inconsistentes, los pods se actualizan automáticamente de acuerdo con la información más reciente de la Deployment.

---

### AVISO

Las Deployments en el estado de actualización deshabilitada no se pueden revertir.

---

- Paso 1** Inicie sesión en la consola de CCE, vaya a un clúster existente y elija **Workloads** en el panel de navegación.
- Paso 2** Haga clic en la ficha **Deployments** y elija **More > Disable/Enable Upgrade** en la columna **Operation** de la carga de trabajo.
- Paso 3** En el cuadro de diálogo que muestra en pantalla, haga clic en **Yes**.

----Fin

## Gestión de etiquetas

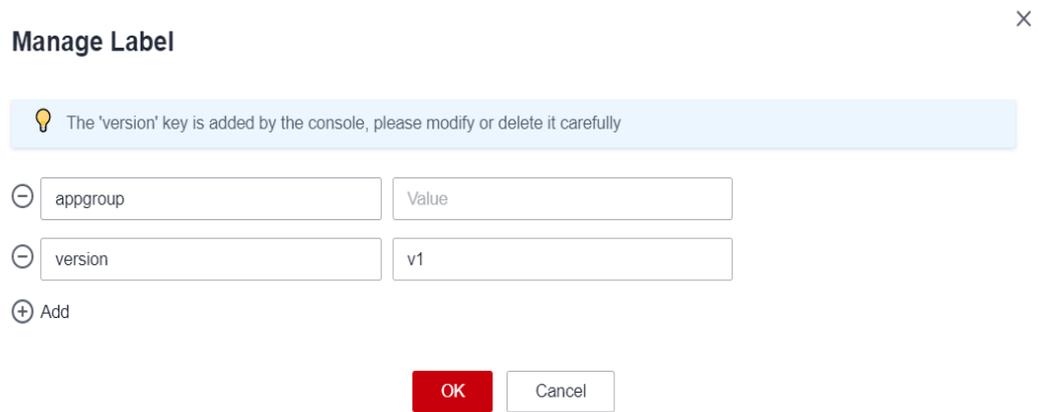
Las etiquetas son pares de clave-valor y se pueden adjuntar a cargas de trabajo. Puede gestionar y seleccionar cargas de trabajo por etiquetas. Puede agregar etiquetas a varias cargas de trabajo o a una carga de trabajo especificada.

- Paso 1** Inicie sesión en la consola de CCE, vaya a un clúster existente y elija **Workloads** en el panel de navegación.

**Paso 2** Haga clic en la ficha **Deployments** y elija **More > Manage Label** en la columna **Operation** de la carga de trabajo de destino.

**Paso 3** Haga clic en **Add**, escriba una clave y un valor y haga clic en **OK**.

**Figura 5-12** Gestión de etiquetas



**NOTA**

Un par de clave-valor debe contener de 1 a 63 caracteres que comiencen y terminen con una letra o un dígito. Solo se permiten letras, dígitos, guiones (-), guiones bajos (\_) y puntos (.).

----Fin

## Supresión de una carga de trabajo/trabajo

Puede eliminar una carga de trabajo o un trabajo que ya no sea necesario. Las cargas de trabajo o trabajos eliminados no se pueden recuperar. Realice esta operación con precaución. En esta sección se utiliza una Deployment como ejemplo para describir cómo eliminar una carga de trabajo.

**Paso 1** Inicie sesión en la consola de CCE, vaya a un clúster existente y elija **Workloads** en el panel de navegación.

**Paso 2** En la misma fila que la carga de trabajo que va a eliminar, elija **Operation > More > Delete**.

Lea las instrucciones del sistema cuidadosamente. Una carga de trabajo no se puede recuperar después de eliminarla. Tenga cuidado al realizar esta operación.

**Paso 3** Haga clic en **Yes**.

**NOTA**

- Si el nodo donde se encuentra el pod no está disponible o se apaga y la carga de trabajo no se puede eliminar, puede eliminar por la fuerza el pod de la lista de pods en la página de detalles de la carga de trabajo.
- Asegúrese de que otras cargas de trabajo no utilizan los volúmenes de almacenamiento que se van a eliminar. Si estos volúmenes se importan o tienen instantáneas, solo puede desvincularlos.

----Fin

## Eventos

En esta sección se utilizan Deployments como ejemplo para ilustrar cómo ver los eventos de una carga de trabajo. Para ver el evento de un trabajo o trabajo cron, haga clic en **View Event** en la columna **Operation** de la carga de trabajo de destino.

**Paso 1** Inicie sesión en la consola de CCE, vaya a un clúster existente y elija **Workloads** en el panel de navegación.

**Paso 2** En la página de ficha **Deployments**, haga clic en la carga de trabajo de destino. En la página de ficha **Pods**, haga clic en **View Events** para ver el nombre del evento, el tipo de evento, el número de ocurrencias, el evento de Kubernetes, la primera hora de ocurrencia y la última hora de ocurrencia.

### NOTA

Los datos del evento se conservarán durante una hora y luego se eliminarán automáticamente.

----Fin

## 5.5 Acceso a un contenedor

### Escenario

Si encuentra problemas inesperados al usar un contenedor puede iniciar sesión en el contenedor para depurarlo.

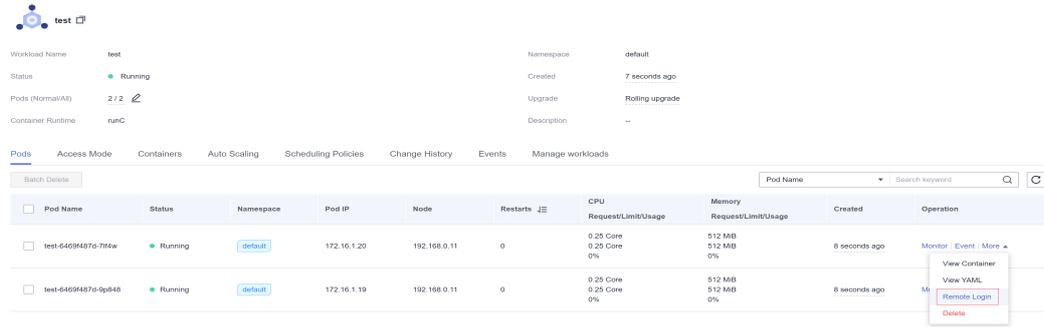
### Inicio de sesión en un contenedor usando CloudShell

#### AVISO

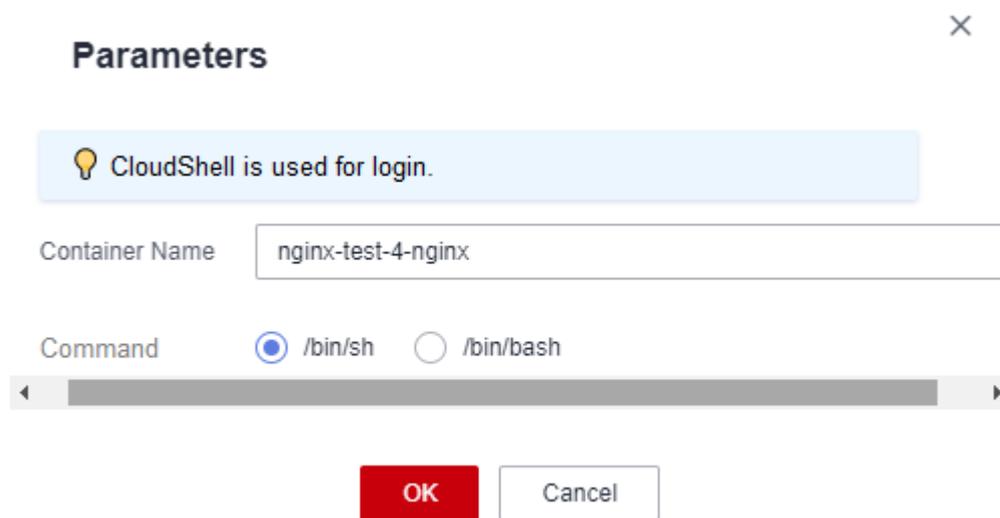
- CloudShell se implementa según VPCEP. Para usar kubectl para acceder a un clúster, debe configurar el grupo de seguridad (*Cluster name-cce-control-Random number*) en el nodo principal del clúster para permitir que los siguientes bloques CIDR accedan al puerto 5443. De forma predeterminada, el puerto 5443 permite el acceso desde todos los bloques CIDR. Si tiene grupos de seguridad reforzados y no se puede acceder a ningún clúster de CloudShell, compruebe si el puerto 5443 permite el acceso desde 198.19.0.0/16.
- Actualmente, puede usar CloudShell para iniciar sesión en contenedores solo en las regiones CN North-Beijing1, CN North-Beijing4, CN East-Shanghai1, CN East-Shanghai2, CN South-Guangzhou y CN North-Ulanqab1.

Puede encontrar la entrada de acceso en la lista de pods de carga de trabajo, como se muestra en la siguiente figura.

**Figura 5-13** Acceso a un contenedor



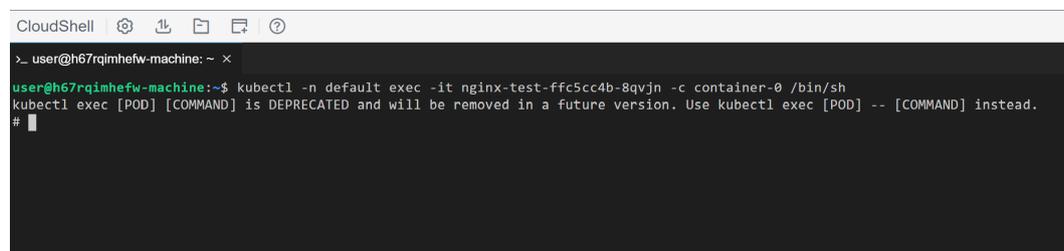
Después de hacer clic en **Remote Login**, se muestra el siguiente cuadro de diálogo. Seleccione el contenedor al que desea acceder y el comando, y haga clic en **OK**.



CloudShell se inicia, kubectl se inicializa y el comando **kubectl exec** se ejecuta automáticamente.

**NOTA**

Espere de 5 a 10 segundos hasta que el comando **kubectl exec** se ejecute automáticamente.



**Inicio de sesión en un contenedor con kubectl**

**Paso 1** Utilice kubectl para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).

**Paso 2** Ejecute el siguiente comando para ver el pod creado:

```
kubectl get pod
```

El resultado del ejemplo es el siguiente:

NAME	READY	STATUS	RESTARTS	AGE
nginx-59d89cb66f-mhljr	1/1	Running	0	11m

**Paso 3** Consulte el nombre del contenedor en el pod.

```
kubectl get po nginx-59d89cb66f-mhljr -o jsonpath='{range .spec.containers[*]}{.name}{end}{"\n"}'
```

El resultado del ejemplo es el siguiente:

```
container-1
```

**Paso 4** Ejecute el siguiente comando para iniciar sesión en el contenedor **container-1** en el pod **nginx-59d89cb66f-mhljr**:

```
kubectl exec -it nginx-59d89cb66f-mhljr -c container-1 -- /bin/sh
```

**Paso 5** Para salir del contenedor, ejecute el comando **exit**.

----Fin

# 6 Planificación

## 6.1 Descripción general

CCE admite diferentes tipos de programación de recursos y programación de tareas, lo que mejora el rendimiento de las aplicaciones y la utilización general de recursos del clúster. Esta sección describe las funciones principales de la programación de recursos de CPU, la programación de recursos heterogéneos de GPU/NPU y la programación de Volcano.

### Programación de CPU

CCE proporciona políticas de CPU para asignar núcleos de CPU físicos completos a las aplicaciones, lo que mejora el rendimiento de las aplicaciones y reduce la latencia de programación de aplicaciones.

Función	Descripción	Documentación
Política de CPU	Cuando muchos pods con uso intensivo de CPU se ejecutan en un nodo, las cargas de trabajo se pueden migrar a diferentes núcleos de CPU. Muchas cargas de trabajo no son sensibles a esta migración y, por lo tanto, funcionan bien sin ninguna intervención. Para las aplicaciones sensibles a la CPU, puede usar la política de CPU proporcionada por Kubernetes para asignar núcleos dedicados a las aplicaciones, mejorando el rendimiento de las aplicaciones y reduciendo la latencia de programación de aplicaciones.	<a href="#">Política de CPU</a>
Política de CPU mejorada	Basado en la política de enlace de núcleo estático de Kubernetes, la política de CPU mejorada (estática mejorada) admite pods explosibles (cuyas solicitudes y límites de CPU son enteros positivos) y permite que ciertas CPUs prioricen estos pods, asegurando la estabilidad de la aplicación.	<a href="#">Política de CPU mejorada</a>

## Programación de GPU

CCE programa recursos de GPU heterogéneos en clústeres y permite que las GPU se utilicen en los contenedores.

Función	Descripción	Documentación
Programación de GPU	La programación de GPU le permite especificar el número de GPU que solicita un pod. El valor puede ser inferior a 1 para que varios pods puedan compartir una GPU.	<a href="#">Programación de GPU</a>

## Programación de NPU

CCE programa recursos de NPU heterogéneos en un clúster para realizar de manera rápida y eficiente la inferencia y el reconocimiento de imágenes.

Función	Descripción	Documentación
Programación de NPU	La programación de NPU le permite especificar el número de NPU que un pod solicita para proporcionar recursos de NPU para las cargas de trabajo.	<a href="#">Programación de NPU</a>

## Programación de volcano

Volcano es una plataforma de procesamiento por lotes basada en Kubernetes que admite aprendizaje automático, aprendizaje profundo, bioinformática, genómica y otras aplicaciones de big data. Proporciona capacidades informáticas de alto rendimiento de propósito general, como la programación de trabajos, la gestión de chips heterogéneos y la gestión de ejecución de trabajos.

Función	Descripción	Documentación
Programador de clústeres predeterminado	Puede configurar Volcano como el planificador predeterminado del clúster. No es necesario especificar manualmente un planificador al crear una carga de trabajo. De esta manera, puede utilizar la capacidad de programación de Volcano más fácilmente.	<a href="#">Configuración del volcano como planificador predeterminado</a>

Función	Descripción	Documentación
Despliegue híbrido de trabajos online y offline	Basado en los tipos de trabajos en línea y fuera de línea, la programación de Volcano se utiliza para utilizar los recursos que se solicitan pero no se utilizan en el clúster (es decir, la diferencia entre el número de recursos solicitados y el número de recursos utilizados), la implementación de sobresuscripción de recursos y despliegue híbridos y la mejora de la utilización de recursos de clúster.	<a href="#">Despliegue híbrido de trabajos en línea y fuera de línea</a>
Programación de afinidad de NUMA	Volcano tiene como objetivo levantar la limitación para hacer que la topología de NUMA del planificador sea consciente de que: <ul style="list-style-type: none"> <li>● Los pods no están programados para los nodos que la topología NUMA no coincide.</li> <li>● Los pods están programados para el mejor nodo para la topología NUMA.</li> </ul>	<a href="#">Programación de afinidad de NUMA</a>

## 6.2 Programación de CPU

### 6.2.1 Política de CPU

#### Escenarios de aplicación

De forma predeterminada, kubelet utiliza [cuotas de CFS](#) para imponer límites de CPU de pod. Cuando el nodo ejecuta muchos pods unidos a CPU, la carga de trabajo puede moverse a diferentes núcleos de CPU dependiendo de si el pod está limitado y qué núcleos de CPU están disponibles en el tiempo de programación. Muchas cargas de trabajo no son sensibles a esta migración y, por lo tanto, funcionan bien sin ninguna intervención. Algunas aplicaciones son sensibles a la CPU. Son sensibles a:

- Estrangulamiento de la CPU
- Cambio de contexto
- Errores de caché del procesador
- Acceso a memoria entre sockets
- Hyperthreads que se espera que se ejecuten en la misma tarjeta de CPU física

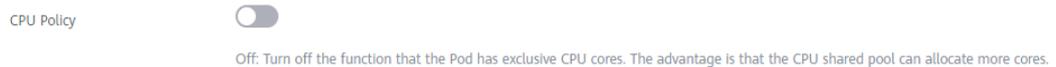
Si sus cargas de trabajo son sensibles a cualquiera de estos elementos y la afinidad de caché de CPU y la latencia de programación afectan significativamente el rendimiento de la carga de trabajo, kubelet permite que las políticas de gestión de CPU alternativas determinen algunas preferencias de ubicación en el nodo. El gestor de CPU asigna preferentemente recursos en un socket y núcleos físicos completos para evitar interferencias.

#### Habilitación de la política de gestión de CPU

Una [política de gestión de CPU](#) es especificada por el indicador kubelet `--cpu-manager-policy`. De forma predeterminada, Kubernetes admite las siguientes políticas:

- Disabled (**none**): la política predeterminada. La política **none** habilita explícitamente el esquema de afinidad de CPU predeterminado existente, sin proporcionar afinidad más allá de lo que el programador del sistema operativo hace automáticamente.
- Enabled (**static**): La política **static** permite que contenedores en pods **Guaranteed** con solicitudes de GPU enteras obtenga una mayor afinidad de CPU y exclusividad en el nodo.

Al crear un clúster, puede configurar la política de gestión de CPU de **Advanced Settings** como se muestra en la siguiente figura.



También puede configurar la política en un grupo de nodos. La configuración cambiará el indicador de kubelet **--cpu-manager-policy** en el nodo. Inicie sesión en la consola de CCE, haga clic en el nombre del clúster, acceda a la página de detalles del clúster y elija **Nodes** en el panel de navegación. En la página mostrada, haga clic en la ficha **Node Pools**. Elija **More > Manage** en la columna **Operation** del grupo de nodos de destino y cambie el valor **decpu-manager-policy** a **static**.

## Permitir que los pods usen exclusivamente los recursos de la CPU

Prerrequisitos:

- Habilite la política **static** en el nodo. Para obtener más información, véase [Habilitación de la política de gestión de CPU](#).
- Tanto **Request** como **Limit** deben establecerse en la definición de pod y sus valores deben ser los mismos.
- Si un contenedor de inicio necesita utilizar exclusivamente la CPU, establezca su **requests** en el mismo que el del contenedor de servicio. De lo contrario, el contenedor de servicio no hereda el resultado de asignación de CPU del contenedor de inicio, y el administrador de CPU reserva más recursos de CPU de los supuestos. Para obtener más información, vea [App Containers can't inherit Init Containers CPUs - CPU Manager Static Policy](#).

Puede usar [Política de programación \(afinidad/antiafinidad\)](#) para programar los pods configurados en los nodos donde está habilitada la política **static**. De esta manera, los pods pueden utilizar exclusivamente los recursos de CPU.

Ejemplo de YAML:

```
kind: Deployment
apiVersion: apps/v1
metadata:
  name: test
spec:
  replicas: 1
  selector:
    matchLabels:
      app: test
  template:
    metadata:
      labels:
        app: test
    spec:
      containers:
        - name: container-1
          image: nginx:alpine
          resources:
```

```

        requests:
          cpu: 2          # The value must be an integer and must be the
same as that in limits.
          memory: 2048Mi
          limits:
            cpu: 2          # The value must be an integer and must be the
same as that in requests.
            memory: 2048Mi
        imagePullSecrets:
          - name: default-secret
    
```

## 6.2.2 Política de CPU mejorada

Kubernetes proporciona dos **políticas de CPU**: ninguna y estática.

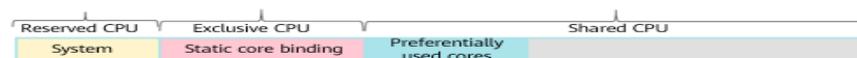
- **none**: La política de CPU está deshabilitada de forma predeterminada, lo que indica el comportamiento de programación existente.
- **static**: La política de enlace estática del núcleo de la CPU está habilitada. Esta política permite que los pods con ciertas características de recursos reciban una afinidad y exclusividad mejoradas de la CPU en el nodo.

Basado en la política estática de Kubernetes, la política de CPU mejorada (estática mejorada) admite pods explosibles (cuyas solicitudes y límites de CPU son enteros positivos) y les permite usar preferentemente ciertas CPUs, asegurando la estabilidad de la aplicación. Por ejemplo:

```

...
spec:
  containers:
  - name: nginx
    image: nginx
    resources:
      limits:
        memory: "300Mi"
        cpu: "2"
      requests:
        memory: "200Mi"
        cpu: "1"
    
```

Esta característica se basa en la programación optimizada de la CPU en el núcleo de Huawei Cloud EulerOS 2.0. Cuando el uso de CPU utilizado preferentemente por un contenedor supera el 85%, el contenedor se asigna automáticamente a otras CPU con bajo uso para garantizar la capacidad de respuesta de las aplicaciones.



### 📖 NOTA

- Cuando se habilita la política de CPU mejorada, el rendimiento de la aplicación es mejor que el de la política **none**, pero peor que el de la política **static**.
- La CPU no sería utilizada exclusivamente por los pods explosibles, todavía está en el grupo de CPU compartido. Cuando los pods explosibles están en la marea baja, otros pods pueden compartir esta CPU.

## Notas y restricciones

Para utilizar esta función, se deben cumplir las siguientes condiciones:

- La versión del clúster es v1.23 o posterior.
- El sistema operativo del nodo es Huawei Cloud EulerOS 2.0.

## Procedimiento

**Paso 1** Inicie sesión en la consola de CCE.

**Paso 2** Haga clic en el nombre del clúster y acceda a la consola del clúster. Elija **Nodes** en el panel de navegación y haga clic en la ficha **Node Pools** de la derecha.

**Paso 3** Seleccione un grupo de nodos cuyo sistema operativo sea **Huawei Cloud EulerOS 2.0** y elija **More > Manage** en la columna **Operation**.



**Paso 4** En la ventana **Manage Component** que se muestra, cambie el valor de **cpu-manager-policy** del componente kubelet a **enhanced-staic**.

### Manage Components (Node Pool [redacted]-nodepool-72676)

Customize the settings of Kubernetes native components or CCE-developed components to satisfy your demands. Details about the parameters: [Configuring Clusters Components](#)

kubelet ^

cpu-manager-policy	<input type="text" value="enhanced-static"/>	Description
kube-api-qps	<input type="text" value="100"/>	
kube-api-burst	<input type="text" value="100"/>	
max-pods	<input type="text" value="20"/>	Description

**Paso 5** Haga clic en **OK**.

----Fin

## Verificación

Tome un nodo con 8 vCPUs y 32 GB de memoria como ejemplo. Despliegue una carga de trabajo cuya solicitud de CPU es 1 y el límite es 2 en el clúster por adelantado.

**Paso 1** Inicie sesión en un nodo en el grupo de nodos y vea la salida **/var/lib/kubelet/cpu\_manager\_state**.

```
cat /var/lib/kubelet/cpu_manager_state
```

Salida del comando:

```
{"policyName":"enhanced-static","defaultCpuSet":"0,2-7","entries":{"6739f6f2-  
ebe5-48ae-945a-986d5d8919b9":{"container-1":"0-7,10001"}}, "checksum":1638128523}
```

- Si el valor de **policyName** es de **enhanced-static**, la política se configura correctamente.

- 10000 se utiliza como base para el ID de CPU. En este ejemplo, 10001 indica que el ID de CPU de afinidad usado por el contenedor es la CPU 1, y 0-7 indica el conjunto de CPU que puede ser usado por el contenedor en el pod.

**Paso 2** Compruebe la configuración de cgroup de `cpuset.preferred_cpus` del contenedor. La salida es el ID de la CPU que se utiliza preferentemente.

```
cat /sys/fs/cgroup/cpuset/kubepods/burstable/pod {pod uid} / {Container ID} / cpuset.preferred_cpus
```

- `{pod uid}` indica el UID de pod, que se puede obtener ejecutando el siguiente comando en el host que se ha conectado al clúster mediante kubectl:

```
kubectl get po {pod name} -n {namespace} -ojsonpath='{.metadata.uid}'
```

En el comando anterior, `{pod name}` y `{namespace}` indican el nombre del pod y el espacio de nombres al que pertenece el pod.

- `{Container id}` debe ser un ID de contenedor completo. Puede ejecutar el siguiente comando en el nodo donde se está ejecutando contenedor para obtener el ID de contenedor:

Grupo de nodos de Docker:

```
docker ps --no-trunc | grep {pod name} | grep -v cce-pause | awk '{print $1}'
```

Grupo de nodos en containerd:

```
crictl ps --no-trunc | grep {pod name} | grep -v cce-pause | awk '{print $1}'
```

Un ejemplo completo es el siguiente:

```
cat /sys/fs/cgroup/cpuset/kubepods/burstable/pod6739f6f2-  
ebe5-48ae-945a-986d5d8919b9/5ba5603434b95fd22d36fba6a5f1c44eba83c18c2e1de9b52ac9b5  
2e93547a13/cpuset.preferred_cpus
```

Si se presenta la siguiente salida de commando, se utiliza preferentemente la CPU 1.

```
1
```

----Fin

## 6.3 Programación de GPU

Puede utilizar GPU en los contenedores de CCE.

### Requisitos previos

- Se ha creado un nodo de GPU. Para obtener más información, véase [Creación de un nodo](#).
- Se ha instalado `gpu-device-plugin` (complemento original de `gpu-beta`). Durante la instalación, seleccione el controlador de GPU en el nodo. Para obtener más información, véase [gpu-device-plugin \(anteriormente gpu-beta\)](#).
- `gpu-device-plugin` monta el directorio del controlador en `/usr/local/nvidia/lib64`. Para usar recursos de GPU en un contenedor debe agregar `/usr/local/nvidia/lib64` a la variable de entorno `LD_LIBRARY_PATH`.

Por lo general, puede utilizar cualquiera de los siguientes métodos para agregar un archivo:

- Configure la variable de entorno `LD_LIBRARY_PATH` en el Dockerfile utilizado para crear una imagen. (Recomendado)
 

```
ENV LD_LIBRARY_PATH /usr/local/nvidia/lib64:$LD_LIBRARY_PATH
```
- Configure la variable de entorno `LD_LIBRARY_PATH` en el comando del inicio de imagen.

```
/bin/bash -c "export LD_LIBRARY_PATH=/usr/local/nvidia/
lib64:$LD_LIBRARY_PATH && ..."
```

- c. Defina la variable de entorno **LD\_LIBRARY\_PATH** al crear una carga de trabajo. (Asegúrese de que esta variable no está configurada en el contenedor. De lo contrario, se sobrescribirá.)

```
env:
  - name: LD_LIBRARY_PATH
    value: /usr/local/nvidia/lib64
```

## Uso de GPU

Cree una carga de trabajo y solicite GPU. Puede especificar el número de GPU de la siguiente manera:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: gpu-test
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      app: gpu-test
  template:
    metadata:
      labels:
        app: gpu-test
    spec:
      containers:
      - image: nginx:perl
        name: container-0
        resources:
          requests:
            cpu: 250m
            memory: 512Mi
            nvidia.com/gpu: 1 # Number of requested GPUs
          limits:
            cpu: 250m
            memory: 512Mi
            nvidia.com/gpu: 1 # Maximum number of GPUs that can be used
        imagePullSecrets:
        - name: default-secret
```

**nvidia.com/gpu** especifica el número de las GPU que se van a solicitar. El valor puede ser inferior a **1**. Por ejemplo, **nvidia.com/gpu: 0.5** indica que varios pods comparten una GPU. En este caso, todos los recursos de GPU solicitados provienen de la misma tarjeta de GPU.

Después de especificar **nvidia.com/gpu**, las cargas de trabajo no se programarán en nodos sin GPU. Si el nodo está exento de GPU, se reportan eventos de Kubernetes similares a los siguientes:

- 0/2 nodes are available: 2 Insufficient nvidia.com/gpu.
- 0/4 nodes are available: 1 InsufficientResourceOnSingleGPU, 3 Insufficient nvidia.com/gpu.

Para usar GPU en la consola de CCE, seleccione la cuota de GPU y especifique el porcentaje de GPU reservado para el contenedor al crear una carga de trabajo.

**Figura 6-1** Uso de GPU

Container

Name

Image Name

CPU Quota Request  cores; Limit  cores

GPU Quota  Do not use

Dedicated

Shared  % All graphics cards

## Etiquetas de nodo de GPU

CCE etiquetará los nodos habilitados para GPU después de que se creen. Diferentes tipos de nodos habilitados para GPU tienen diferentes etiquetas.

```
$ kubectl get node -L accelerator
NAME                STATUS    ROLES    AGE    ACCELERATOR
VERSION
10.100.2.179        Ready    <none>   8m43s  v1.19.10-r0-
CCE21.11.1.B006-21.11.1.B006  nvidia-t4
```

Al utilizar las GPU, puede habilitar la afinidad entre los pods y los nodos en función de las etiquetas para que los pods se puedan programar en los nodos correctos.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: gpu-test
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      app: gpu-test
  template:
    metadata:
      labels:
        app: gpu-test
    spec:
      nodeSelector:
        accelerator: nvidia-t4
      containers:
        - image: nginx:perl
          name: container-0
      resources:
        requests:
          cpu: 250m
```

```
memory: 512Mi
nvidia.com/gpu: 1 # Number of requested GPUs
limits:
  cpu: 250m
  memory: 512Mi
  nvidia.com/gpu: 1 # Maximum number of GPUs that can be used
imagePullSecrets:
- name: default-secret
```

## 6.4 Programación de NPU

Puede utilizar NPU en los contenedores de CCE.

### Requisitos previos

- Se ha creado un nodo de NPU. Para obtener más información, véase [Creación de un nodo](#).
- El huawei-npu ha sido instalado. Para obtener más información, véase [huawei-npu](#).

### Uso de NPU

Cree una carga de trabajo y solicite NPU. Puede especificar el número de NPU de la siguiente manera:

```
kind: Deployment
apiVersion: apps/v1
metadata:
  name: npu-test
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      app: npu-test
  template:
    metadata:
      labels:
        app: npu-test
    spec:
      containers:
      - name: container-0
        image: nginx:perl
        resources:
          limits:
            cpu: 250m
            huawei.com/ascend-310: '1'
            memory: 512Mi
          requests:
            cpu: 250m
            huawei.com/ascend-310: '1'
            memory: 512Mi
        imagePullSecrets:
        - name: default-secret
```

Especifica el número de NPU que se van a solicitar en **huawei.com/ascend-310**.

Después de especificar **huawei.com/ascend-310**, las cargas de trabajo no se programarán en nodos sin NPU. Si las NPU son insuficientes, se informará de un evento de Kubernetes similar a "0/2 nodes are available: 2 Insufficient huawei.com/ascend-310".

Para utilizar las NPU en la consola de CCE, seleccione la cuota de Ascend 310 y especifique el número de chips de Ascend que se utilizarán al crear una carga de trabajo.

**Figura 6-2** Uso de NPU

from  
YAML

Pull Policy  Always ?

Image Tag --Select--

Memory Request  MIB.Limit  MIB

Quota

NPU Quota  Use

Number of Ascend 310 chips required by the container. The value must be an integer.

## Etiquetas de nodo de NPU

CCE etiquetará los nodos habilitados para NPU que están listos para usar.

```
$ kubectl get node -L accelerator/huawei-npu
NAME                STATUS    ROLES    AGE
VERSION            HUAWEI-NPU
10.100.2.59        Ready    <none>   2m18s   v1.19.10-r0-
CCE21.11.1.B006-21.11.1.B006   ascend-310
```

Al utilizar las NPU, puede habilitar la afinidad entre los pods y los nodos en función de las etiquetas para que los pods se puedan programar en los nodos correctos.

```
kind: Deployment
apiVersion: apps/v1
metadata:
  name: npu-test
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      app: npu-test
  template:
    metadata:
      labels:
        app: npu-test
    spec:
      nodeSelector:
        accelerator/huawei-npu: ascend-310
      containers:
        - name: container-0
          image: nginx:perl
          resources:
            limits:
              cpu: 250m
              huawei.com/ascend-310: '1'
              memory: 512Mi
            requests:
              cpu: 250m
              huawei.com/ascend-310: '1'
              memory: 512Mi
          imagePullSecrets:
            - name: default-secret
```

## 6.5 Programación de volcano

## 6.5.1 Configuración del volcano como planificador predeterminado

Volcano es una plataforma de procesamiento por lotes basada en Kubernetes que admite aprendizaje automático, aprendizaje profundo, bioinformática, genómica y otras aplicaciones de big data. Proporciona capacidades informáticas de alto rendimiento de propósito general, como la programación de trabajos, la gestión de chips heterogéneos y la gestión de ejecución de trabajos.

Para utilizar el programador de volcano, debe configurar manualmente **schedulerName** en la carga de trabajo en **volcano**. Para obtener más información acerca de cómo especificar un planificador para una carga de trabajo, vea [Configuración de varios planificadores](#).

Esta sección describe cómo configurar volcano como el planificador predeterminado para un clúster sin cambiar el nombre del planificador para usar la capacidad de programación de volcano más fácilmente.

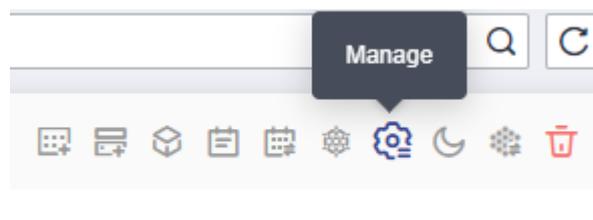
### Cambio del planificador predeterminado de un clúster existente

#### 📖 NOTA

- Solo los clústeres de v1.19.16 o posterior admiten esta función.
- El complemento de volcano de v1.6.0 o posterior debe estar instalado.
- Si el complemento de volcano de 1.6.0 o una versión posterior se ha instalado en un clúster de una versión anterior y se ha actualizado a v1.19.16 o una versión posterior, debe actualizar o reinstalar el complemento de volcano para habilitar la capacidad de configuración predeterminada del planificador.
  - Incluso cuando volcano se establece como el programador predeterminado, el controlador de volcano todavía necesita ser programado por kube-scheduler. Por lo tanto, cuando se instala o actualiza el complemento de volcano de una versión adecuada, se especifica un nombre al planificador volcano para la programación de kube-scheduler. Puede ver el valor de **schedulerName** en el archivo YAML. Si el valor es de **kube-scheduler**, volcano es el planificador predeterminado. Si el valor es de **default-scheduler**, actualice o reinstale el complemento de volcano.
  - Actualización in situ: En la página **Add-ons**, seleccione un complemento y haga clic en **Edit**. No es necesario modificar ningún parámetro. Haga clic en **OK**.
  - Reinstalación: En la página **Add-ons**, desinstale el complemento de volcano y vuelva a instalarlo. Tenga en cuenta que cuando se desinstala el complemento, los CRD de volcano, como vcjob, se eliminan para evitar el impacto en los servicios.
- Cada vez que se conmuta el modo planificador de un grupo de inventario, los componentes volcano y kube-scheduler se reinician para conmutar el modo planificador. Las excepciones de programación pueden ocurrir durante un corto período de tiempo durante la conmutación.

**Paso 1** Inicie sesión en la consola de CCE.

**Paso 2** En la página de lista de clústeres, haga clic en el icono de gestión del clúster de destino.



**Paso 3** Si la versión actual del clúster admite el modo de planificador predeterminado especificado, el parámetro **default-scheduler** se muestra en el elemento de configuración **kube-scheduler** para establecer el modo de planificador actual del clúster.

- **kube-scheduler**: programador predeterminado de un clúster nativo de Kubernetes.
- **volcano**: programador de volcanes mejorado. Cuando se utiliza este programador, el complemento de **volcano** debe estar instalado en el clúster.

Ajusta **default-scheduler** a **volcano**.



**Paso 4** Una vez completada la configuración, haga clic en **OK**.

----Fin

## 6.5.2 Despliegue híbrido de trabajos en línea y fuera de línea

### Trabajos en línea y fuera de línea

Los trabajos se pueden clasificar en trabajos en línea y trabajos fuera de línea en función de si los servicios están siempre en línea.

- **Online job**: Estos trabajos se ejecutan durante mucho tiempo, con aumentos de tráfico regulares, solicitudes de recursos de marea y altos requisitos en SLA, como servicios de publicidad y comercio electrónico.
- **Offline jobs**: Estos trabajos se ejecutan durante poco tiempo, tienen altos requisitos de cómputo y pueden tolerar una alta latencia, como la IA y los servicios de big data.

### Sobresuscripción de recursos y despliegue híbrido

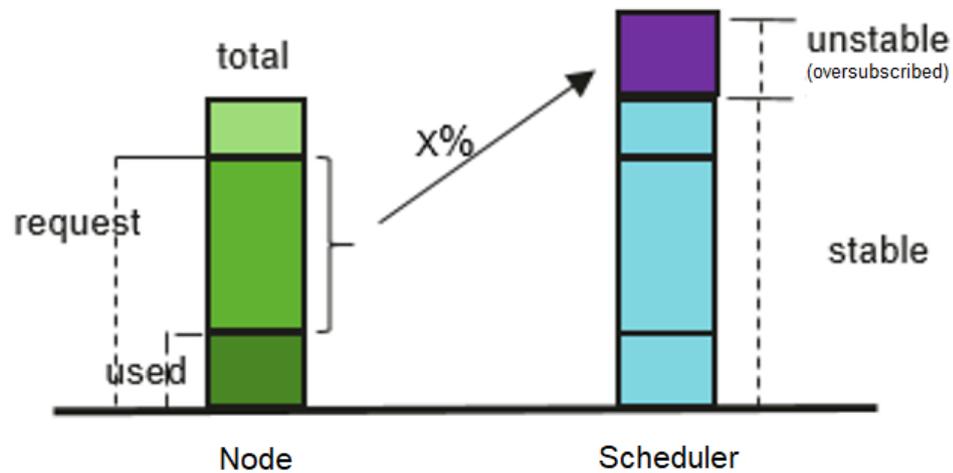
Muchos servicios ven aumentos en el tráfico. Para garantizar el rendimiento y la estabilidad, a menudo se solicitan recursos al máximo necesario. Sin embargo, las oleadas pueden disminuir muy pronto y los recursos, si no se liberan, se desperdician en horas no pico. Especialmente para trabajos en línea que solicitan una gran cantidad de recursos para garantizar el SLA, la utilización de recursos puede ser tan baja como sea posible.

La sobresuscripción de recursos es el proceso de hacer uso de los recursos solicitados inactivos. Los recursos sobresuscritos son adecuados para desplegar trabajos sin conexión, que se centran en el rendimiento, pero tienen bajos requisitos de SLA y pueden tolerar ciertos fallos.

El despliegue híbrido de trabajos en línea y sin conexión en un clúster pueden utilizar mejor los recursos del clúster.

**Figura 6-3** Sobresuscripción de recursos

$$\text{Oversubscription} = (\text{request} - \text{used}) \times \text{Ratio}$$



## Sobresuscripción para despliegue híbrido

Se admite el despliegue híbrido, y los recursos de CPU y memoria se pueden sobresuscribir. Las características clave son las siguientes:

- Los trabajos sin conexión se ejecutan preferentemente en los nodos sobresuscritos.  
 Si existen los nodos sobresuscritos y no sobresuscritos, el primero puntuará más alto que el último y los trabajos fuera de línea se programan preferentemente para los nodos sobresuscritos.
- Los trabajos en línea solo pueden usar recursos que no estén sobresuscritos si se programan para un nodo sobresuscrito.  
 Los trabajos sin conexión pueden utilizar recursos tanto sobresuscritos como no sobresuscritos de un nodo sobresuscrito.
- En el mismo período de programación, los trabajos en línea tienen prioridad sobre los trabajos sin conexión.  
 Si existen trabajos en línea y sin conexión, los trabajos en línea se programan primero. Cuando el uso de recursos de nodo excede el límite superior y las solicitudes de nodo superan el 100%, se desalojarán los trabajos sin conexión.
- El aislamiento de CPU/memoria es proporcionado por los núcleos.  
 Aislamiento de CPU: Los trabajos en línea pueden adelantarse rápidamente a los recursos de CPU de los trabajos sin conexión y suprimir el uso de CPU de los trabajos sin conexión.  
 Aislamiento de memoria: cuando se agotan los recursos de memoria del sistema y se activa OOM Kill, el núcleo desaloja primero los trabajos sin conexión.
- Reglas de admisión de trabajos sin conexión de kubelet:  
 Después de programar el pod en un nodo, kubelet inicia el pod solo cuando los recursos del nodo pueden cumplir con la solicitud de pod (predicateAdmitHandler.Admit). kubelet inicia el pod cuando se cumplen las dos condiciones siguientes:

- La solicitud total de pods que se iniciarán y los trabajos en ejecución en línea < nodos asignables
- La solicitud total de pods que se iniciarán y el trabajo en ejecución online/offline < nodos asignables+nodos sobresuscritos
- Sobresuscripción de recursos y despliegue híbrido:

Si solo se utiliza el despliegue híbrido, debe configurar la etiqueta **volcano.sh/colocation=true** para el nodo y eliminar la etiqueta de nodo **volcano.sh/oversubscription** o establecer su valor en **false**.

Si la etiqueta **volcano.sh/colocation=true** está configurada para un nodo, el despliegue híbrido está habilitado. Si la etiqueta **volcano.sh/oversubscription=true** está configurada, la sobresuscripción de recursos está habilitada. En la siguiente tabla se enumeran las combinaciones de funciones disponibles después de activar el despliegue híbrido o la sobresuscripción de recursos.

Despliegue híbrido activado (volcano.sh/colocation=true)	Sobresuscripción de recursos activada(volcano.sh/oversubscription=true)	¿Usar recursos sobresuscritos?	Condiciones para desalojar los pod fuera de línea
No	No	No	No hay
Sí	No	No	El uso de recursos de nodo excede el umbral alto.
No	Sí	Sí	El uso de recursos de nodo excede el umbral alto, y la solicitud de nodo excede el 100%.
Sí	Sí	Sí	El uso de recursos de nodo excede el umbral alto.

## Notas y restricciones

### Especificaciones

- Versión del clúster
  - v1.19: v1.19.16-r4 o posterior
  - v1.21: v1.21.7-r0 o posterior
  - v1.23: v1.23.5-r0 o posterior
  - v1.25 o posterior
- Tipo de clúster: CCE o CCE Turbo
- Sistema operativo del nodo: EulerOS 2.9 (kernel-4.18.0-147.5.1.6.h729.6.eulerosv2r9.x86\_64) o Huawei Cloud EulerOS 2.0
- Tipo de nodo: ECS

- La versión adicional del volcán: 1.7.0 o posterior

### Restricciones

- Antes de activar el complemento de sobresuscripción del volcán, asegúrese de que el complemento de sobresuscripción no esté activado.
- La modificación de la etiqueta de un nodo sobresuscrito no afecta a los pods en ejecución.
- Los pods en ejecución no se pueden convertir entre servicios en línea y sin conexión. Para convertir los servicios, necesita reconstruir los pods.
- Si la etiqueta **volcano.sh/oversubscription=true** está configurada para un nodo del clúster, la configuración **oversubscription** debe agregarse al complemento volcán. De lo contrario, la planificación de los nodos sobrevendidos será anormal. Asegúrese de que ha configurado correctamente las etiquetas porque el planificador no comprueba las configuraciones de complementos y nodos. Para obtener más información sobre las etiquetas, consulte [Configuración de etiquetas de sobresuscripción para la programación](#).
- Para deshabilitar la sobresuscripción, realice las siguientes operaciones:
  - Quite la etiqueta **volcano.sh/oversubscription** del nodo sobresuscrito.
  - Establezca **over-subscription-resource** a **false**.
  - Modifique el mapa de configuración del planificador del volcán llamado **volcano-scheduler-configmap** y quite el complemento de sobresuscripción.
- Si **cpu-manager-policy** se establece en enlace de núcleo estático en un nodo, no asigne la clase QoS de Guaranteed a los pods sin conexión. Si se requiere la unión del núcleo, cambie los pods a los en línea. De lo contrario, los pods sin conexión pueden ocupar la CPU de los pods en línea, causando errores de inicio de pods en línea, y los pods sin conexión no se pueden iniciar aunque se programan correctamente.
- Si **cpu-manager-policy** se establece como enlace de núcleo estático en un nodo, no enlaza núcleos a todos los pods en línea. De lo contrario, los pods en línea ocupan todos los recursos de CPU o memoria, dejando un pequeño número de recursos sobresuscritos.

## Configuración de etiquetas de sobresuscripción para la programación

Si la etiqueta **volcano.sh/oversubscription=true** está configurada para un nodo del clúster, la configuración **oversubscription** debe agregarse al complemento volcán. De lo contrario, la planificación de los nodos sobrevendidos será anormal. Para obtener más información sobre la configuración relacionada, consulte [Tabla 6-1](#).

Asegúrese de que ha configurado correctamente las etiquetas porque el planificador no comprueba las configuraciones de complementos y nodos.

**Tabla 6-1** Configuración de etiquetas de sobresuscripción para la programación

Sobresuscripción en complementos	Etiqueta de sobresuscripción en el nodo	Planificación
Sí	Sí	Activado por sobresuscripción
Sí	No	Activado
No	No	Activado

Sobrescripción en complementos	Etiqueta de sobrescripción en el nodo	Planificación
No	Sí	No activado o fallado. Evite esta configuración.

## Uso del despliegue híbrido

### Paso 1 Configurar el complemento de volcano.

1. Utilice kubectl para conectarse al clúster.
2. Instale el complemento de volcado y agregue el complemento **oversubscription** a **volcano-scheduler-configmap**. Asegúrese de que la configuración del complemento no contenga el complemento **overcommit**. Si **- name: overcommit** existe, elimine esta configuración.

```
# kubectl edit cm volcano-scheduler-configmap -n kube-system
apiVersion: v1
data:
  volcano-scheduler.conf: |
    actions: "enqueue, allocate, backfill"
    tiers:
    - plugins:
      - name: gang
      - name: priority
      - name: conformance
      - name: oversubscription
    - plugins:
      - name: drf
      - name: predicates
      - name: nodeorder
      - name: binpack
    - plugins:
      - name: cce-gpu-topology-predicate
      - name: cce-gpu-topology-priority
      - name: cce-gpu
```

### Paso 2 Habilite la función de sobrescripción de nodo.

Una etiqueta se puede configurar para usar recursos sobrescritos solo después de que la función sobrescripción esté habilitada para un nodo. Los nodos relacionados solo se pueden crear en un grupo de nodos. Para habilitar la función de sobrescripción, realice los siguientes pasos:

1. Cree un grupo de nodos.
2. Elija **More > Manage** en la columna **Operation** del grupo de nodos creado.
3. En la ventana **Manage Component** que se muestra, establezca **over-subscription-resource** en **kubelet** en **true** y haga clic en **OK**.

kubelet ^

cpu-manager-policy	none
kube-api-qps	100
kube-api-burst	100
max-pods	110
pod-pids-limit	-1
with-local-dns	false
event-qps	5
allowed-unsafe-sysctls	[]
over-subscription-resource	true

**Paso 3** Establezca la etiqueta de sobresuscripción del nodo.

La etiqueta **volcano.sh/oversubscription** debe configurarse para un nodo sobresuscrito. Si esta etiqueta se establece para un nodo y el valor es **true**, el nodo es un nodo sobresuscrito. De lo contrario, el nodo no es un nodo sobresuscrito.

```
kubectl label node 192.168.0.0 volcano.sh/oversubscription=true
```

Un nodo con sobresuscripción también admite los umbrales de sobresuscripción, como se indica en **Tabla 6-2**. Por ejemplo:

```
kubectl annotate node 192.168.0.0 volcano.sh/evicting-cpu-high-watermark=70
```

Consultar la información del nodo

```
# kubectl describe node 192.168.0.0
Name:          192.168.0.0
Roles:         <none>
Labels:        ...
               volcano.sh/oversubscription=true
Annotations:   ...
               volcano.sh/evicting-cpu-high-watermark: 70
```

**Tabla 6-2** Anotaciones de sobresuscripción de nodos

Nombre	Descripción
volcano.sh/evicting-cpu-high-watermark	<p>Cuando el uso de CPU de un nodo excede el valor especificado, se activa el desalojo de trabajo sin conexión y el nodo se vuelve impredecible.</p> <p>El valor predeterminado es <b>80</b>, que indica que el desalojo de trabajos sin conexión se activa cuando el uso de CPU de un nodo supera el 80%.</p>

Nombre	Descripción
volcano.sh/evicting-cpu-low-watermark	Después de activar el desalojo, la programación comienza de nuevo cuando el uso de CPU de un nodo es menor que el valor especificado. El valor predeterminado es <b>30</b> , que indica que la programación comienza de nuevo cuando el uso de CPU de un nodo es inferior al 30%.
volcano.sh/evicting-memory-high-watermark	Cuando el uso de memoria de un nodo excede el valor especificado, se activa el desalojo del trabajo sin conexión y el nodo se vuelve impredecible. El valor predeterminado es <b>60</b> , que indica que el desalojo del trabajo sin conexión se activa cuando el uso de memoria de un nodo supera el 60%.
volcano.sh/evicting-memory-low-watermark	Después de activar el desalojo, la programación comienza de nuevo cuando el uso de memoria de un nodo es menor que el valor especificado. El valor por defecto es el <b>30</b> que indica que la programación comienza de nuevo cuando el uso de memoria de un nodo es inferior al 30%.
volcano.sh/oversubscription-types	Tipo de recurso sobresuscrito. Las opciones son las siguientes: <ul style="list-style-type: none"> <li>● CPU (CPU con sobresuscripción)</li> <li>● memoria (memoria sobresuscrita)</li> <li>● cpu,memoria (sobresuscritas CPU y memoria)</li> </ul> El valor predeterminado es <b>cpu,memory</b> .

**Paso 4** Despliegue los trabajos en línea y fuera de línea.

La etiqueta **volcano.sh/qos-level** debe agregarse a la anotación para distinguir los trabajos sin conexión. El valor es un entero que oscila entre -7 y 7. Si el valor es menor que 0, el trabajo es un trabajo sin conexión. Si el valor es mayor o igual que 0, el trabajo es un trabajo de alta prioridad, es decir, un trabajo en línea. No es necesario establecer esta etiqueta para los trabajos en línea. Para los trabajos en línea y fuera de línea, establezca **schedulerName** en **volcano** para habilitar el programador Volcano.

 **NOTA**

Las prioridades de los trabajos online/online y offline/online no se diferencian, y la validez del valor no se verifica. Si el valor de **volcano.sh/qos-level** de un trabajo sin conexión no es un entero negativo que oscila entre -7 y 0, el trabajo se procesa como un trabajo en línea.

Para un trabajo fuera de línea:

```
kind: Deployment
apiVersion: apps/v1
spec:
  replicas: 4
  template:
    metadata:
      annotations:
        metrics.alpha.kubernetes.io/custom-endpoints:
' [{"api":"","path":"","port":"","names":""}]'
        volcano.sh/qos-level: "-1" # Offline job label
    spec:
      schedulerName: volcano # The Volcano scheduler is used.
      ...
```

Para un trabajo en línea:

```
kind: Deployment
apiVersion: apps/v1
spec:
  replicas: 4
  template:
    metadata:
      annotations:
        metrics.alpha.kubernetes.io/custom-endpoints:
' [{"api":"","path":"","port":"","names":""}]'
    spec:
      schedulerName: volcano # The Volcano scheduler is used.
      ...
```

**Paso 5** Ejecute el siguiente comando para comprobar el número de recursos sobresuscritos y el uso de recursos:

kubectl describe el nodo <nodeIP>

```
# kubectl describe node 192.168.0.0
Name:          192.168.0.0
Roles:         <none>
Labels:        ...
               volcano.sh/oversubscription=true
Annotations:   ...
               volcano.sh/oversubscription-cpu: 2335
               volcano.sh/oversubscription-memory: 341753856
Allocatable:
  cpu:          3920m
  memory:       6263988Ki
Allocated resources:
 (Total limits may be over 100 percent, i.e., overcommitted.)
Resource       Requests      Limits
-----
cpu             4950m (126%)  4950m (126%)
memory         1712Mi (27%)  1712Mi (27%)
```

----Fin

## Ejemplo de despliegue híbrido

A continuación se utiliza un ejemplo para describir cómo desplegar trabajos en línea y sin conexión en modo híbrido.

**Paso 1** Suponga que un clúster tiene dos nodos: un nodo sobresuscrito y un nodo no sobresuscrito.

```
# kubectl get node
NAME           STATUS    ROLES    AGE    VERSION
192.168.0.173  Ready    <none>   4h58m  v1.19.16-r2-CCE22.5.1
192.168.0.3    Ready    <none>   148m   v1.19.16-r2-CCE22.5.1
```

- 192.168.0.173 es un nodo sobresuscrito (con la etiqueta **volcano.sh/oversubscription=true**).

- 192.168.0.3 es un nodo no sobresuscrito (sin la etiqueta **volcano.sh/oversubscription=true**).

```
# kubectl describe node 192.168.0.173
Name:          192.168.0.173
Roles:         <none>
Labels:        beta.kubernetes.io/arch=amd64
               ...
               volcano.sh/oversubscription=true
```

**Paso 2** Enviar solicitudes de creación de trabajos sin conexión. Si los recursos son suficientes, todos los trabajos sin conexión se programarán en el nodo sobresuscrito.

La plantilla de trabajo sin conexión es la siguiente:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: offline
  namespace: default
spec:
  replicas: 2
  selector:
    matchLabels:
      app: offline
  template:
    metadata:
      labels:
        app: offline
      annotations:
        volcano.sh/qos-level: "-1"          # Offline job label
    spec:
      schedulerName: volcano                # The Volcano scheduler is used.
      containers:
        - name: container-1
          image: nginx:latest
          imagePullPolicy: IfNotPresent
          resources:
            requests:
              cpu: 500m
              memory: 512Mi
            limits:
              cpu: "1"
              memory: 512Mi
          imagePullSecrets:
            - name: default-secret
```

Los trabajos sin conexión se programan en el nodo sobresuscrito.

```
# kubectl get pod -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP              NODE
offline-69cdd49bf4-pmjp8            1/1    Running   0          5s    192.168.10.178 192.168.0.173
offline-69cdd49bf4-z8kxh            1/1    Running   0          5s    192.168.10.131 192.168.0.173
```

**Paso 3** Envíe solicitudes de creación de empleo en línea. Si los recursos son suficientes, los trabajos en línea se programarán para el nodo no sobresuscrito.

La plantilla de trabajo en línea es la siguiente:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: online
  namespace: default
spec:
  replicas: 2
  selector:
    matchLabels:
      app: online
  template:
```

```

metadata:
  labels:
    app: online
spec:
  schedulerName: volcano # The Volcano scheduler is used.
  containers:
  - name: container-1
    image: resource_consumer:latest
    imagePullPolicy: IfNotPresent
    resources:
      requests:
        cpu: 1400m
        memory: 512Mi
      limits:
        cpu: "2"
        memory: 512Mi
    imagePullSecrets:
    - name: default-secret
    
```

Los trabajos en línea se programan para el nodo no sobresuscrito.

```

# kubectl get pod -o wide
NAME                READY   STATUS    RESTARTS   AGE   IP           NODE
online-ffb46f656-4mwr6  1/1    Running   0           5s   192.168.10.146  192.168.0.3
online-ffb46f656-dqdv2  1/1    Running   0           5s   192.168.10.67   192.168.0.3
    
```

**Paso 4** Mejore el uso de recursos del nodo sobresuscrito y observe si se activa el desalojo de trabajos sin conexión.

Despliegue trabajos en línea en el nodo sobresuscrito (192.168.0.173).

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: online
  namespace: default
spec:
  replicas: 2
  selector:
    matchLabels:
      app: online
  template:
    metadata:
      labels:
        app: online
    spec:
      affinity: # Submit an online job to an
      nodeAffinity:
        requiredDuringSchedulingIgnoredDuringExecution:
          nodeSelectorTerms:
            - matchExpressions:
                - key: kubernetes.io/hostname
                  operator: In
                  values:
                    - 192.168.0.173
      schedulerName: volcano # The Volcano scheduler is used.
      containers:
      - name: container-1
        image: resource_consumer:latest
        imagePullPolicy: IfNotPresent
        resources:
          requests:
            cpu: 700m
            memory: 512Mi
          limits:
            cpu: 700m
            memory: 512Mi
    
```

```
imagePullSecrets:
  - name: default-secret
```

Envíe los trabajos en línea o sin conexión al nodo sobresuscrito (192.168.0.173) al mismo tiempo.

```
# kubectl get pod -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP              NODE
offline-69cdd49bf4-pmjp8           1/1    Running   0          13m   192.168.10.178 192.168.0.173
offline-69cdd49bf4-z8kxh           1/1    Running   0          13m   192.168.10.131 192.168.0.173
online-6f44bb68bd-b8z9p            1/1    Running   0          3m4s  192.168.10.18  192.168.0.173
online-6f44bb68bd-g6xk8            1/1    Running   0          3m12s 192.168.10.69  192.168.0.173
```

Observe el nodo sobresuscrito (192.168.0.173). Puede encontrar que existen recursos sobresuscritos y que la tasa de asignación de CPU supera el 100%.

```
# kubectl describe node 192.168.0.173
Name:                               192.168.0.173
Roles:                               <none>
Labels:                               ...
                                     volcano.sh/oversubscription=true
Annotations:                          ...
                                     volcano.sh/oversubscription-cpu: 2343
                                     volcano.sh/oversubscription-memory: 3073653200
                                     ...
Allocated resources:
 (Total limits may be over 100 percent, i.e., overcommitted.)
Resource           Requests          Limits
-----
cpu                 4750m (121%)     7350m (187%)
memory             3760Mi (61%)     4660Mi (76%)
...

```

Aumente el uso de la CPU de los trabajos en línea en el nodo. Se activa el desalojo de trabajo sin conexión.

```
# kubectl get pod -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP              NODE
offline-69cdd49bf4-bwdm7           1/1    Running   0          11m   192.168.10.208 192.168.0.3
offline-69cdd49bf4-pmjp8           0/1    Evicted   0          26m   <none>          192.168.0.173
offline-69cdd49bf4-qpds            1/1    Running   0          11m   192.168.10.174 192.168.0.3
offline-69cdd49bf4-z8kxh           0/1    Evicted   0          26m   <none>          192.168.0.173
online-6f44bb68bd-b8z9p            1/1    Running   0          24m   192.168.10.18  192.168.0.173
online-6f44bb68bd-g6xk8            1/1    Running   0          24m   192.168.10.69  192.168.0.173
```

----Fin

## Sugerencias sobre el manejo

- Después de reiniciar el kubelet del nodo sobresuscrito, la vista de recursos del planificador del volcano no se sincroniza con la de kubelet. Como resultado, OutOfCPU se produce en algunos trabajos programados recientemente, lo cual es normal. Después de un período de tiempo, el programador de Volcano puede programar correctamente trabajos en línea y fuera de línea.
- Después de enviar trabajos en línea y fuera de línea, no se recomienda cambiar dinámicamente el tipo de trabajo (agregando o eliminando anotación `_volcano.sh/qos-level: "-1"`) porque el núcleo actual no admite el cambio de un trabajo fuera de línea a un trabajo en línea.

- CCE recopila el uso de recursos (CPU/memoria) de todos los pods que se ejecutan en un nodo basado en la información de estado en el sistema cgroups. El uso de recursos puede ser diferente del uso de recursos monitorizado, por ejemplo, las estadísticas de recursos mostradas ejecutando el comando **top**.
- Puede agregar recursos sobresuscritos (como CPU y memoria) en cualquier momento. Puede reducir los tipos de recursos sobresuscritos solo cuando la tasa de asignación de recursos no exceda del 100%.

## 6.5.3 Programación de afinidad de NUMA

### Antecedente

Cuando el nodo ejecuta muchos pods unidos a CPU, la carga de trabajo puede moverse a diferentes núcleos de CPU dependiendo de si el pod está limitado y qué núcleos de CPU están disponibles en el tiempo de programación. Muchas cargas de trabajo no son sensibles a esta migración y, por lo tanto, funcionan bien sin ninguna intervención. Sin embargo, en cargas de trabajo en las que la afinidad de la memoria caché de la CPU y la latencia de programación afectan significativamente el rendimiento de la carga de trabajo, el kubelet permite políticas alternativas de gestión de la CPU para determinar algunas preferencias de ubicación en el nodo.

Tanto el CPU Manager como el Topology Manager son componentes kubelet, pero tienen las siguientes limitaciones:

- El planificador no es consciente de la topología. Por lo tanto, la carga de trabajo puede planificarse en un nodo y, a continuación, fallar en el nodo debido al Topology Manager. Esto es inaceptable para los puestos de trabajo TensorFlow. Si algún trabajador o ps falla en el nodo, el trabajo fallará.
- Los administradores son a nivel de nodo que resulta en una incapacidad para hacer coincidir el mejor nodo para la topología de NUMA en todo el clúster.

Para obtener más información, consulte <https://github.com/volcano-sh/volcano/blob/master/docs/design/numa-aware.md>.

Volcano tiene como objetivo resolver la limitación de hacer que la topología de NUMA del planificador sea consciente para lograr lo siguiente:

- No programe pods para los nodos que no coincidan con la topología de NUMA.
- Programe los pods en el mejor nodo para la topología de NUMA.

### Ámbito de aplicación

- Compatible con la programación de topología de recursos de CPU
- Compatible con las políticas de topología a nivel de pod

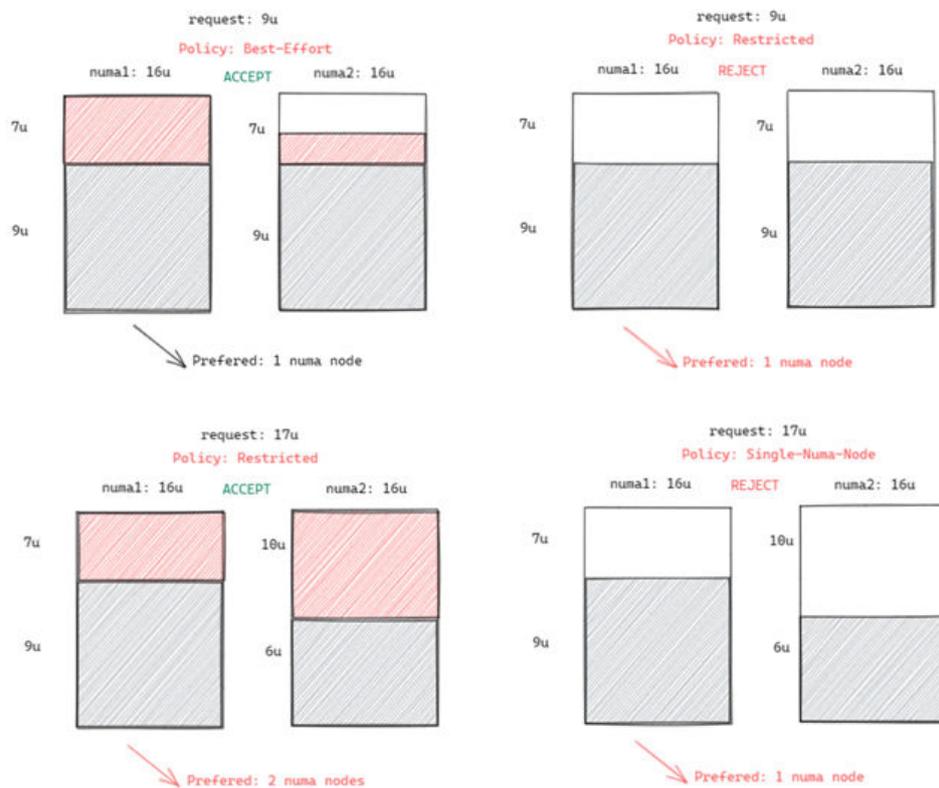
### Predicción de programación

Para los pods con la política de topología, predique la lista de nodos coincidentes.

política	acción
Ninguna	1. Sin acción de filtro

política	acción
best-effort	1. Filtre el nodo con el <b>best-effort</b> de política de topología.
restricted	1. Filtre el nodo con el <b>restricted</b> de política de topología. 2. Filtrar el nodo que la topología de la CPU cumple con los requisitos de la CPU para <b>restricted</b> .
single-numa-node	1. Filtre el nodo con el <b>single-numa-node</b> de política de topología. 2. Filtrar el nodo que la topología de la CPU cumple con los requisitos de la CPU para <b>single-numa-node</b> .

Figura 6-4 Comparación de políticas de programación de NUMA



## Planificación de prioridades

La política de topología tiene como objetivo programar pods para el nodo óptimo. En este ejemplo, cada nodo se puntúa para ordenar el nodo óptimo.

Principio: Programe pods para los nodos de trabajo que requieren el número mínimo de nodos de NUMA.

La fórmula de puntuación es la siguiente:

$$\text{score} = \text{weight} * (100 - 100 * \text{numaNodeNum} / \text{maxNumaNodeNum})$$

Descripción de parámetros:

- **weight**: indica la ponderación del NUMA Aware Plugin.
- **numaNodeNum**: indica el número de nodos de NUMA necesarios para ejecutar el pod en el nodo de trabajo.
- **maxNumaNodeNum**: indica el número máximo de nodos de NUMA en un pod de todos los nodos de trabajo.

## Habilitar volcano para apoyar la programación de afinidad de NUMA

**Paso 1** Habilite la política de gestión de CPU. Para obtener más información, véase [Habilitación de la política de gestión de CPU](#).

**Paso 2** Configure una política de topología de CPU.

1. Inicie sesión en la consola de CCE y acceda a la consola del clúster. Elija **Nodos** en el panel de navegación. En el panel derecho, haga clic en la ficha **Node Pools**. Elija **More > Manage** en la columna **Operation** del grupo de nodos de destino.
2. Cambie el valor de **topology-manager-policy** en **kubelet** a la política de topología de CPU requerida. Como se muestra en la siguiente figura, la política de topología de la CPU es **best-effort**.

Las políticas de topología válidas son **none**, **best-effort**, **restricted** y **single-numa-node**. Para obtener más información sobre estas políticas, consulte [Predicción de programación](#).



**Paso 3** El parámetro **resource\_exporter\_enable** está habilitado para que el complemento de volcano recopile información de NUMA del nodo.

```
{
  "plugins": {
    "eas_service": {
      "availability_zone_id": "",
      "driver_id": "",
      "enable": "false",
      "endpoint": "",
      "flavor_id": "",
      "network_type": "",
      "network_virtual_subnet_id": "",
      "pool_id": "",
      "project_id": "",
      "secret_name": "eas-service-secret"
    }
  },
  "resource_exporter_enable": "true"
}
```

Una vez activada esta función, puede ver la información de topología de NUMA del nodo actual.

```
kubectl get numatopo
NAME      AGE
node-1   4h8m
node-2   4h8m
node-3   4h8m
```

**Paso 4** Habilite el complemento del algoritmo de reconocimiento de numa del volcano.

```
kubectl edit cm -n kube-system volcano-scheduler-configmap
kind: ConfigMap
apiVersion: v1
```

```

metadata:
  name: volcano-scheduler-configmap
  namespace: kube-system
data:
  default-scheduler.conf: |-
    actions: "allocate, backfill"
    tiers:
    - plugins:
      - name: priority
      - name: gang
      - name: conformance
    - plugins:
      - name: overcommit
      - name: drf
      - name: predicates
      - name: nodeorder
    - plugins:
      - name: cce-gpu-topology-predicate
      - name: cce-gpu-topology-priority
      - name: cce-gpu
    - plugins:
      - name: nodelocalvolume
      - name: nodeemptydirvolume
      - name: nodeCSIScheduling
      - name: networkresource
      arguments:
        NetworkType: vpc-router
    - name: numa-aware # add it to enable numa-aware plugin
      arguments:
        weight: 10 # the weight of the NUMA Aware Plugin
    
```

----Fin

## Usar volcano para apoyar la programación de afinidad de NUMA

**Paso 1** Configurar la afinidad de NUMA para las Deployments. A continuación se presenta un ejemplo:

```

kind: Deployment
apiVersion: apps/v1
metadata:
  name: numa-tset
spec:
  replicas: 1
  selector:
    matchLabels:
      app: numa-tset
  template:
    metadata:
      labels:
        app: numa-tset
      annotations:
        volcano.sh/numa-topology-policy: single-numa-node # set the topology
policy
spec:
  containers:
    - name: container-1
      image: nginx:alpine
      resources:
        requests:
          cpu: 2 # The value must be an integer and must be the
same as that in limits.
          memory: 2048Mi
        limits:
          cpu: 2 # The value must be an integer and must be the
same as that in requests.
          memory: 2048Mi
      imagePullSecrets:
        - name: default-secret
    
```

**Paso 2** Cree un trabajo de volcano y use la afinidad de NUMA.

```
apiVersion: batch.volcano.sh/v1alpha1
kind: Job
metadata:
  name: vj-test
spec:
  schedulerName: volcano
  minAvailable: 1
  tasks:
    - replicas: 1
      name: "test"
      topologyPolicy: best-effort # set the topology policy for task
      template:
        spec:
          containers:
            - image: alpine
              command: ["/bin/sh", "-c", "sleep 1000"]
              imagePullPolicy: IfNotPresent
              name: running
              resources:
                limits:
                  cpu: 20
                  memory: "100Mi"
              restartPolicy: OnFailure
```

**Paso 3** Compruebe el uso de NUMA.

```
# Check the CPU usage of the current node.
lscpu
...
CPU(s):          32
NUMA node(s):   2
NUMA node0 CPU(s): 0-15
NUMA node1 CPU(s): 16-31

# Check the CPU allocation of the current node.
cat /var/lib/kubelet/cpu_manager_state
{"policyName":"static","defaultCpuSet":"0,10-15,25-31","entries":{"777870b5-
c64f-42f5-9296-688b9dc212ba":{"container-1":"16-24"},"fb15e10a-
b6a5-4aaa-8fcd-76c1aa64e6fd":{"container-1":"1-9"},"checksum":318470969}
```

----Fin

# 7 Red

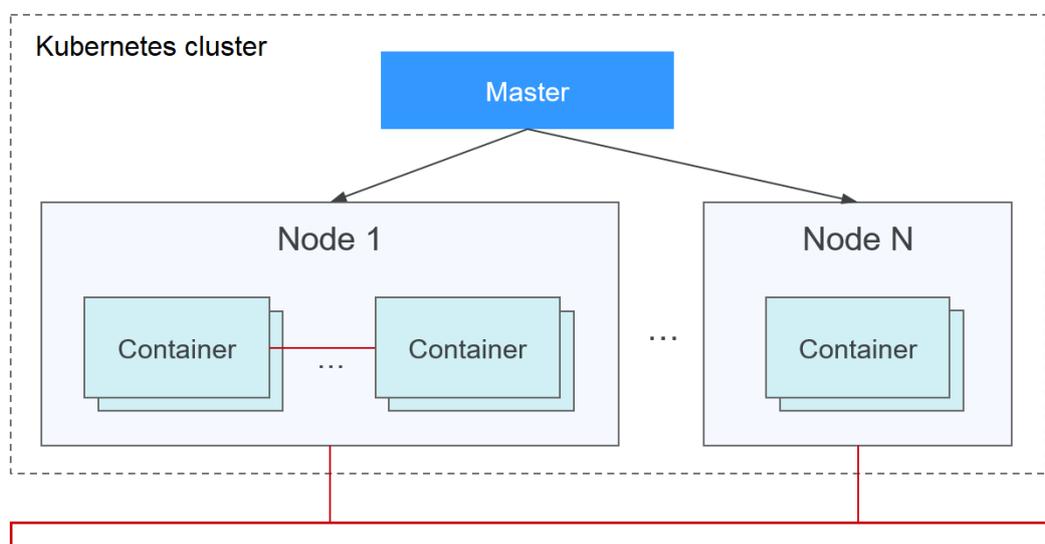
## 7.1 Descripción general

Puede obtener información sobre una red de clústeres de los dos aspectos siguientes:

- ¿Cómo es una red de clústeres? Un clúster consta de varios nodos, y los pods (o contenedores) se ejecutan en los nodos. Los nodos y los contenedores necesitan comunicarse entre sí. Para obtener más información acerca de los tipos de red del clúster y sus funciones, consulte [Estructura de red de clústeres](#).
- ¿Cómo se implementa el acceso a pod en un clúster? El acceso a un pod o contenedor es un proceso de acceso a los servicios de un usuario. Kubernetes proporciona [Service](#) y [Entrada](#) para solucionar problemas de acceso a pods. Esta sección resume los escenarios comunes de acceso a la red. Puede seleccionar el escenario adecuado en función de los requisitos del sitio. Para obtener más información sobre los escenarios de acceso a la red, consulte [Escenarios de acceso](#).

### Estructura de red de clústeres

Todos los nodos del clúster se encuentran en una VPC y utilizan la red VPC. La red de contenedor es gestionada por complementos de red dedicados.



- **Red de nodo**

Una red de nodos asigna direcciones IP a hosts (nodos en la figura anterior) en un clúster. Debe seleccionar una subred de VPC como red de nodo del clúster de CCE. El número de direcciones IP disponibles en una subred determina el número máximo de nodos (incluidos los nodos maestros y los nodos de trabajo) que se pueden crear en un clúster. Esta cantidad también se ve afectada por la red de contenedor. Para obtener más información, consulte el modelo de red de contenedor.

- **Red de contenedor**

Una red de contenedor asigna direcciones IP a contenedores en un clúster. CCE hereda el modelo de red IP-Per-Pod-Per-Network de Kubernetes. Es decir, cada pod tiene una dirección IP independiente en un plano de red y todos los contenedores en un pod comparten el mismo espacio de nombres de red. Todos los pods de un clúster existen en una red plana conectada directamente. Pueden acceder entre sí con sus direcciones IP sin usar NAT. Kubernetes solo proporciona un mecanismo de red para los pods, pero no configura directamente las redes de pods. La configuración de redes pod se implementa mediante complementos de la red de contenedor específicos. Los complementos de red de contenedor son responsables de configurar las redes para los pods y gestionar las direcciones IP de contenedor.

Actualmente, CCE es compatible con los siguientes modelos de la red de contenedor:

- Red de túneles de contenedores: La red de túneles de contenedor se construye sobre la red de nodos, pero es independiente de ella, con la encapsulación de túneles. Este modelo de red utiliza VXLAN para encapsular paquetes de Ethernet en paquetes de UDP y los transmite en túneles. Open vSwitch sirve como el conmutador virtual de back-end.
- Red de VPC: La red de VPC utiliza el enrutamiento de VPC para integrarse con la red subyacente. Este modelo de red es aplicable a escenarios de alto rendimiento. El número máximo de nodos permitidos en un clúster depende de la cuota de ruta en una red de VPC. A cada nodo se le asigna un bloque CIDR de un tamaño fijo. Este modelo de red está libre de sobrecarga de encapsulación de túnel y supera el modelo de red de túnel contenedor. Además, como el enrutamiento de VPC incluye rutas a direcciones IP de nodo y el bloque CIDR de contenedor, se puede acceder directamente a los pods de contenedor en un clúster desde fuera del clúster.
- Desarrollado por CCE, Cloud Native Network 2.0 integra profundamente las interfaces de red elástica (ENI) y las interfaces de subred (sub-ENI) de VPC. Las direcciones IP del contenedor se asignan desde el bloque CIDR de VPC. Se admite la red de paso a través de ELB para las solicitudes de acceso directo a contenedores. Los grupos de seguridad y las IP elásticas (EIP) están destinados a ofrecer un alto rendimiento.

El rendimiento, la escala de red y los escenarios de aplicación de una red contenedor varían según el modelo de red contenedor. Para obtener más información sobre las funciones y características de los diferentes modelos de la red de contenedor, consulte [Descripción general](#).

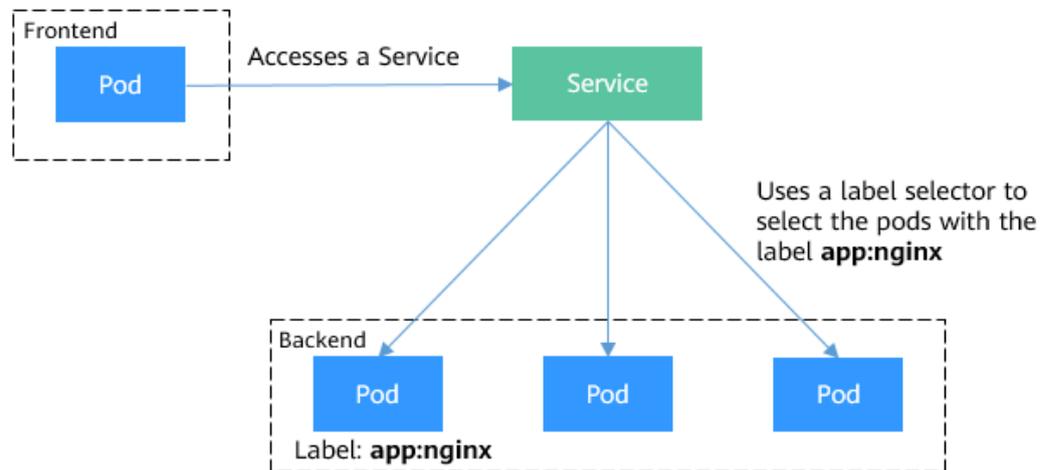
- **Red de Service**

El Service también es un objeto de Kubernetes. Cada Service tiene una dirección IP fija. Al crear un clúster en CCE, puede especificar el bloque CIDR de Service. El bloque CIDR de Service no puede superponerse con el bloque CIDR de nodo o contenedor. El bloque CIDR de Service solo se puede usar dentro de un clúster.

## Service

Se utiliza un Service para el acceso a pods. Con una dirección IP fija, un Service reenvía el tráfico de acceso a los pods y realiza el balanceo de carga para estos pods.

**Figura 7-1** Acceso a pods con un Service



Puede configurar los siguientes tipos de servicios:

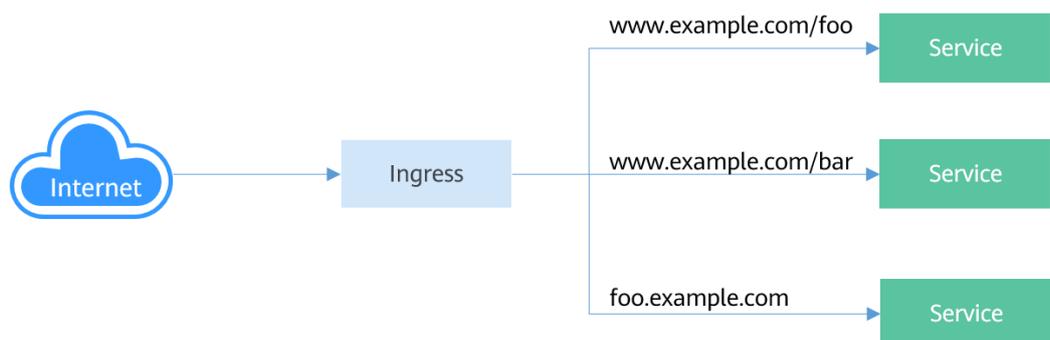
- ClusterIP: se utiliza para hacer que el Service solo sea accesible desde un clúster.
- NodePort: se utiliza para el acceso desde fuera de un clúster. Se accede a un Service de NodePort a través del puerto del nodo.
- LoadBalancer: se utiliza para el acceso desde fuera de un clúster. Es una extensión de NodePort a la que un balanceador de carga se dirige, y los sistemas externos solo necesitan acceder al balanceador de carga.
- DNAT: se utiliza para el acceso desde fuera de un clúster. Traduce direcciones para nodos de clúster y permite que varios nodos de clúster compartan una EIP.

Para obtener más información sobre el Service, consulte [Descripción general](#).

## Entrada

Services reenvía solicitudes con protocolos TCP y UDP de capa 4. Ingress reenvía solicitudes con protocolos HTTP y HTTPS de capa 7. Los nombres de dominio y las rutas se pueden usar para lograr detalles más finos.

**Figura 7-2** Ingress y Service



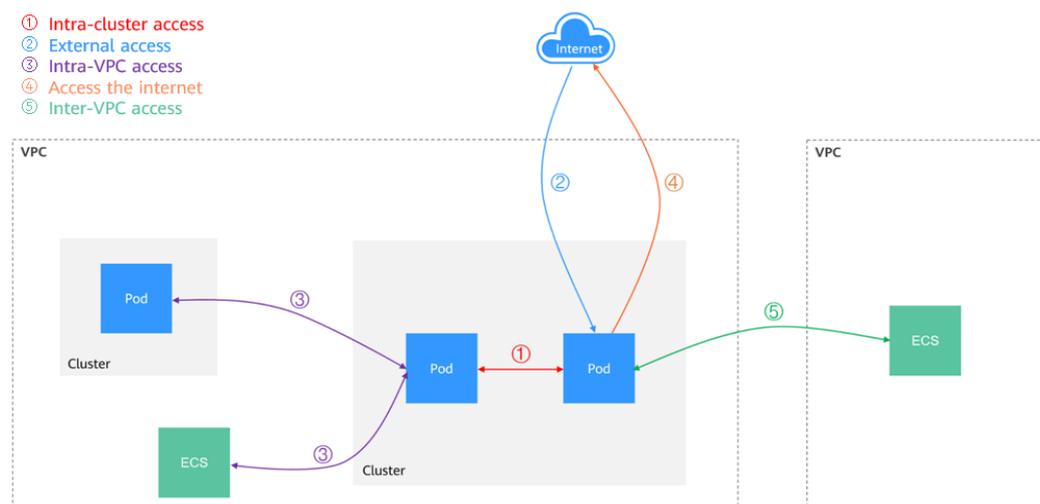
Para obtener más información sobre la entrada, consulte [Descripción de entrada](#).

## Escenarios de acceso

Los escenarios de acceso a la carga de trabajo se pueden clasificar de la siguiente manera:

- Acceso dentro del clúster: se utiliza un Service de ClusterIP para que las cargas de trabajo del mismo clúster tengan acceso entre sí.
- Acceso desde fuera de un clúster: se recomienda un Service (tipo NodePort o LoadBalancer) o un ingreso para una carga de trabajo fuera de un clúster para acceder a las cargas de trabajo del clúster.
  - Acceso con la red pública: Una EIP debe estar vinculada al nodo o al balanceador de carga.
  - Acceso con la red privada: Se puede acceder a la carga de trabajo con la dirección IP interna del nodo o balanceador de carga. Si las cargas de trabajo se encuentran en diferentes VPC, se requiere una interconexión para permitir la comunicación entre diferentes VPC.
- La carga de trabajo puede acceder a la red externa de la siguiente manera:
  - Acceso a una intranet: la carga de trabajo accede a la dirección de intranet, pero el método de implementación varía según los modelos de la red de contenedor. Asegúrese de que el grupo de seguridad del par permita las solicitudes de acceso desde el bloque CIDR contenedor. Para más detalles, consulte [Configuración del acceso dentro de la VPC](#).
  - Acceso a una red pública: Es necesario asignar una EIP al nodo donde se ejecuta la carga de trabajo (cuando se utiliza el modelo de red de VPC o de túnel), vincular una EIP a la dirección IP del pod (cuando se utiliza el modelo de red nativa de nube 2.0) o configurar reglas de SNAT con el gateway de NAT. Para obtener más información, véase [Acceso a redes públicas desde un contenedor](#).

**Figura 7-3** Diagrama de acceso a la red



## 7.2 Container Network Models

## 7.2.1 Descripción general

La red de contenedor asigna las direcciones IP a los pods de un clúster y proporciona servicios de red. En CCE, puede seleccionar los siguientes modelos de red para su clúster:

- [Red de túneles](#)
- [Red de VPC](#)
- [Cloud Native Network 2.0](#)

### Comparación de modelos de red

**Tabla 7-1** describe las diferencias de los modelos de red soportados por CCE.



Después de crear un clúster, no se puede cambiar el modelo de red.

**Tabla 7-1** Comparación de modelos de red

Dimensión	Red de túneles	Red de VPC	Cloud Native Network 2.0
Escenarios de aplicación	<ul style="list-style-type: none"> <li>● Escenarios comunes del servicio de contenedor</li> <li>● Escenarios que no tienen altos requisitos de latencia y ancho de banda de la red</li> </ul>	<ul style="list-style-type: none"> <li>● Escenarios que tienen altos requisitos de latencia y ancho de banda de la red</li> <li>● Los contenedores pueden comunicarse con las máquinas virtuales mediante un marco de registro de microservicios, como Dubbo y CSE.</li> </ul>	<ul style="list-style-type: none"> <li>● Escenarios que tienen altos requisitos de latencia, ancho de banda y rendimiento de la red</li> <li>● Los contenedores pueden comunicarse con las máquinas virtuales mediante un marco de registro de microservicios, como Dubbo y CSE.</li> </ul>
Tecnología básica	OVS	IPvlan y ruta de VPC	VPC ENI/sub-ENI
Clústeres aplicables	Clúster de CCE	Clúster de CCE	Clúster de Turbo de CCE
Aislamiento de red	NetworkPolicy nativa de Kubernetes para pods	No	Los pods admiten el aislamiento de grupos de seguridad.
Redes passthrough	No	No	Sí

Dimensión	Red de túneles	Red de VPC	Cloud Native Network 2.0
Gestión de direcciones IP	<ul style="list-style-type: none"> <li>● El bloque CIDR de contenedor se asigna por separado.</li> <li>● Los bloques CIDR se dividen por nodo y pueden asignarse dinámicamente (los bloques CIDR pueden agregarse dinámicamente después de asignarse)</li> </ul>	<ul style="list-style-type: none"> <li>● El bloque CIDR de contenedor se asigna por separado.</li> <li>● Los bloques CIDR se dividen por nodo y se asignan estáticamente (el bloque CIDR no se puede cambiar después de crear un nodo).</li> </ul>	El bloque CIDR de contenedor se divide de la subred de VPC y no necesita asignarse por separado.
rendimiento de la red	Pérdida de rendimiento debido a la encapsulación de VXLAN	Sin encapsulación de túnel. Los paquetes de nodo cruzado se reenvían con routers de VPC, lo que ofrece un rendimiento equivalente al de la red host.	La red de contenedor está integrada con la red de VPC, eliminando la pérdida de rendimiento.
Ajuste de red	Se admite un máximo de 2,000 nodos.	De forma predeterminada, se admiten 200 nodos.  Cada vez que se agrega un nodo al clúster, se agrega una ruta a las tablas de rutas de VPC (incluidas las predeterminadas y las personalizadas). Por lo tanto, el ajuste de clúster está limitada por las tablas de ruta de VPC.	Se admite un máximo de 2,000 nodos.

### AVISO

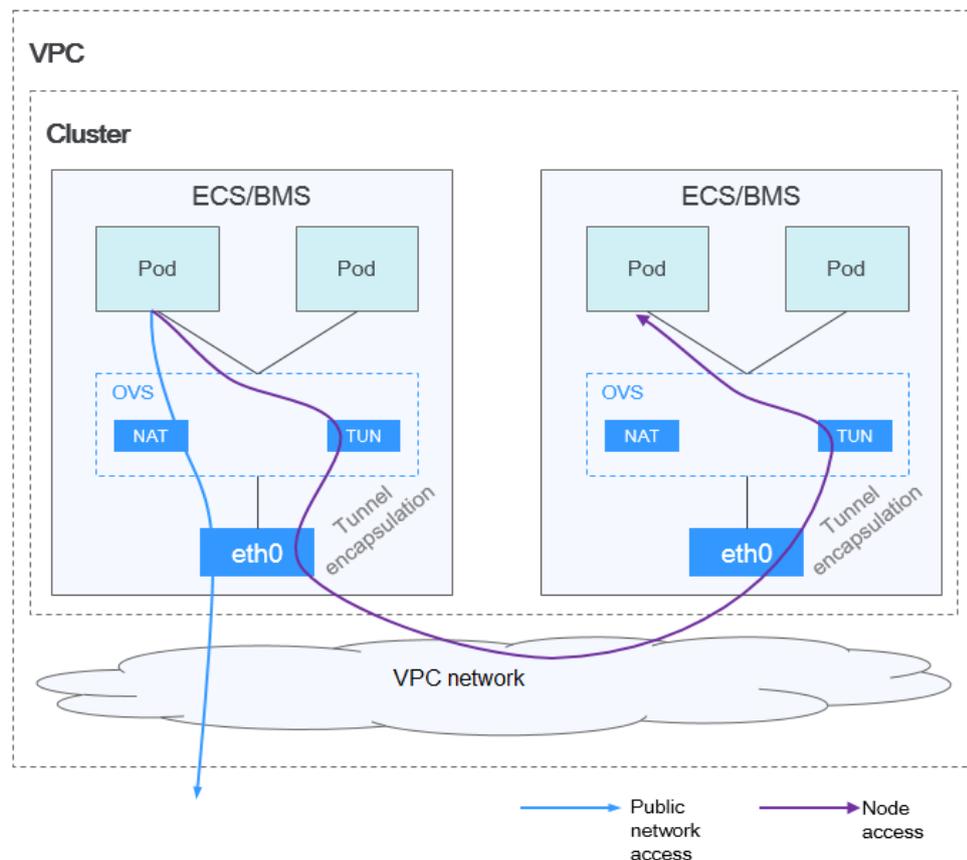
1. El ajuste de un clúster que utiliza el modelo de red de VPC está limitada por las rutas personalizadas de la VPC. Por lo tanto, debe estimar el número de nodos necesarios antes de crear un clúster.
2. El ajuste de un clúster que utiliza el modelo de Cloud Native Network 2.0 depende del tamaño del bloque CIDR de subred de VPC seleccionado para la definición de adjuntos de red. Antes de crear un clúster, evalúe el ajuste del clúster.
3. De forma predeterminada, la red de enrutamiento de VPC admite la comunicación directa entre contenedores y hosts en la misma VPC. Si se configura una política de interconexión entre la VPC y otra VPC, los contenedores puede comunicarse directamente con los hosts en la VPC del mismo nivel. Además, en escenarios de redes híbridas como Direct Connect y VPN, la comunicación entre contenedores y hosts en el extremo del peer también se puede lograr con una planificación adecuada.
4. No cambie la máscara del bloque CIDR primario en la VPC después de crear un clúster. De lo contrario, la red será anormal.

## 7.2.2 Red de túneles de contenedores

### Modelo de red de túneles de contenedores

La red de túneles de contenedor se construye sobre la red de nodos, pero es independiente de ella, con la encapsulación de túneles. Este modelo de red utiliza VXLAN para encapsular paquetes de Ethernet en paquetes de UDP y los transmite en túneles. Open vSwitch sirve como el conmutador virtual de back-end. Aunque a algunos costes de rendimiento, la encapsulación de paquetes y la transmisión de túneles permiten una mayor interoperabilidad y compatibilidad con características avanzadas (como el aislamiento basado en políticas de red) para los escenarios más comunes.

Figura 7-4 Red de túneles de contenedores



### Comunicación de pod a pod

- En el mismo nodo: los paquetes se reenvían directamente a través del puente de OVS en el nodo.
- Entre los nodos: Los paquetes se encapsulan en el puente de OVS y luego se reenvían al nodo del otro extremo.

## Ventajas y desventajas

### Ventajas

- La red de contenedor está desacoplada de la red de nodos y no está limitada por las cuotas de VPC y la velocidad de respuesta (como el número de rutas de VPC, el número de ENI elásticos y la velocidad de creación).
- Se admite el aislamiento de red. Para obtener más información, véase [Network Policies](#).
- Se admiten límites de ancho de banda.
- Se admite la creación de redes a gran escala.

### Desventajas

- Sobrecarga alta de encapsulación, redes complejas y rendimiento bajo
- Error al utilizar las capacidades de balanceo de carga y grupo de seguridad proporcionadas por la VPC
- Las redes externas no se pueden conectar directamente a las direcciones IP de contenedor.

## Escenarios posibles

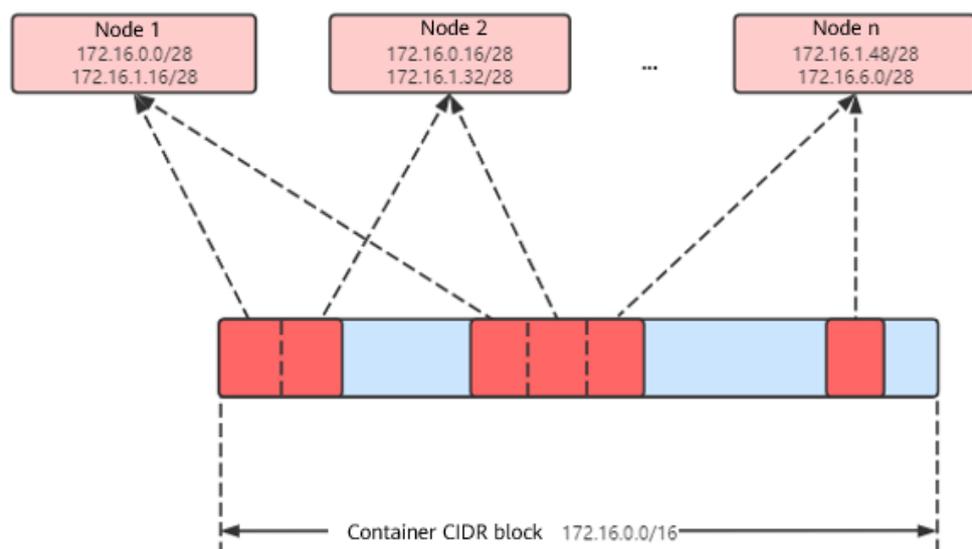
- Bajos requisitos de rendimiento: Como la red de túnel contenedor requiere una encapsulación adicional de túnel VXLAN, tiene entre un 5% y un 15% de pérdida de rendimiento en comparación con los otros dos modelos de red contenedor. Por lo tanto, la red de túnel contenedor es aplicable a escenarios que no tienen requisitos de alto rendimiento, como aplicaciones web y servicios de extremo medio y de extremo medio con un pequeño número de solicitudes de acceso.
- Redes a gran escala: A diferencia de la red de VPC que está limitada por la cuota de ruta de VPC, la red de túnel contenedor no tiene ninguna restricción en la infraestructura. Además, la red de túnel de contenedor controla el dominio de difusión al nivel de nodo. La red de túneles de contenedor admite un máximo de 2000 nodos.

## Gestión de direcciones IP de contenedores

La red de túneles contenedor asigna las direcciones IP de contenedor de acuerdo con las siguientes reglas:

- El bloque CIDR de contenedor se asigna por separado, lo que es irrelevante para el bloque CIDR de nodo.
- Las direcciones IP se asignan por nodo. Uno o más bloques CIDR con un tamaño fijo (16 por defecto) se asignan a cada nodo en un grupo desde el bloque CIDR de contenedor.
- Cuando se agotan las direcciones IP de un nodo, puede solicitar un nuevo bloque CIDR.
- El bloque CIDR de contenedor asigna cíclicamente bloques CIDR a los nuevos nodos o los nodos existentes en secuencia.
- Los pods programados para un nodo son direcciones IP asignadas cíclicamente desde uno o más bloques CIDR asignados al nodo.

Figura 7-5 Asignación de direcciones IP de la red de túneles de contenedor



Número máximo de nodos que se pueden crear en el clúster usando la red de túnel de contenedor = Número de direcciones IP en el bloque CIDR de contenedor / Tamaño del

bloque CIDR de IP asignado al nodo por el bloque CIDR de contenedor a la vez (16 por defecto)

Por ejemplo, si el bloque CIDR de contenedor es 172.16.0.0/16, el número de direcciones IP es 65536. Si se asignan 16 direcciones IP a un nodo a la vez, se puede crear un máximo de 4096 (65536/16) nodos en el clúster. Este es un caso extremo. Si se crean 4096 nodos, se puede crear un máximo de 16 pods para cada nodo porque solo se asignan 16 bloques CIDR IP's a cada nodo. Además, el número de nodos que se pueden crear en un clúster también depende de la red de nodos y de la escala del clúster.

**Figura 7-6** Selección de un modelo de red (al crear el clúster)

**Network Settings** Select the VPC and CIDR blocks for creating nodes and containers in the cluster.

Network Model VPC network **Tunnel network** [? Network Model Overview](#)

Model used for container networking in a cluster. Not editable after creation

VPC --Select-- [Create VPC](#)

CIDR block used by master nodes and worker nodes in the cluster. Not editable after creation

Container CIDR Block **Manually set** Auto select [? How to plan CIDR blocks?](#)

10 . 0 . 0 . 0 / 16

**Max. pods supported by the current networking configuration: 65,533; Max. nodes: 4,096**

## Recomendación para la planificación de bloques CIDR

Como se describe en [Estructura de red de clústeres](#), las direcciones de red de un clúster se pueden dividir en tres partes: red de nodo, red de contenedor y red de servicio. Al planificar direcciones de red, tenga en cuenta los siguientes aspectos:

- Los tres bloques CIDR no pueden superponerse. De lo contrario, se produce un conflicto. Todas las subredes (incluidas las creadas a partir del bloque CIDR secundario) en la VPC donde reside el clúster no pueden entrar en conflicto con los bloques CIDR de contenedor y de Service.
- Asegúrese de que cada bloque CIDR tenga suficientes direcciones IP.
  - Las direcciones IP en el bloque CIDR del nodo deben coincidir con la escala del clúster. De lo contrario, no se pueden crear nodos debido a la insuficiencia de direcciones IP.
  - Las direcciones IP en el bloque CIDR de contenedor deben coincidir con la escala de servicio. De lo contrario, los pods no se pueden crear debido a la insuficiencia de direcciones IP. El número de pods que se pueden crear en cada nodo también depende de otros parámetros. Para más detalles, consulte [Número máximo de pods que se pueden crear en un nodo](#).

## Ejemplo de acceso a la red de túneles de contenedores

Cree un clúster que utilice el modelo de red de túnel de contenedor. Cree una Deployment en el clúster.

```
kind: Deployment
apiVersion: apps/v1
```

```

metadata:
  name: example
  namespace: default
spec:
  replicas: 4
  selector:
    matchLabels:
      app: example
  template:
    metadata:
      labels:
        app: example
    spec:
      containers:
      - name: container-0
        image: 'nginx:perl'
        resources:
          limits:
            cpu: 250m
            memory: 512Mi
          requests:
            cpu: 250m
            memory: 512Mi
      imagePullSecrets:
      - name: default-secret
    
```

Vea el pod creado.

```

$ kubectl get pod -owide
NAME                                READY   STATUS    RESTARTS   AGE   IP
NODE                                NOMINATED NODE   READINESS GATES
example-5bdc5699b7-5rvq4            1/1     Running   0           3m28s  10.0.0.20
192.168.0.42 <none>                 <none>
example-5bdc5699b7-984j9            1/1     Running   0           3m28s  10.0.0.21
192.168.0.42 <none>                 <none>
example-5bdc5699b7-1fxkm            1/1     Running   0           3m28s  10.0.0.22
192.168.0.42 <none>                 <none>
example-5bdc5699b7-wjcmg            1/1     Running   0           3m28s  10.0.0.52
192.168.0.64 <none>                 <none>
    
```

En este caso, no se puede acceder directamente a la dirección IP del pod fuera del clúster en la misma VPC. Esta es una característica de la red de túneles de contenedor.

Sin embargo, se puede acceder al pod desde un nodo en el clúster o en el pod. Como se muestra en la siguiente figura, se puede acceder al pod directamente desde el contenedor.

```

$ kubectl exec -it example-5bdc5699b7-5rvq4 -- curl 10.0.0.21
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
    
```

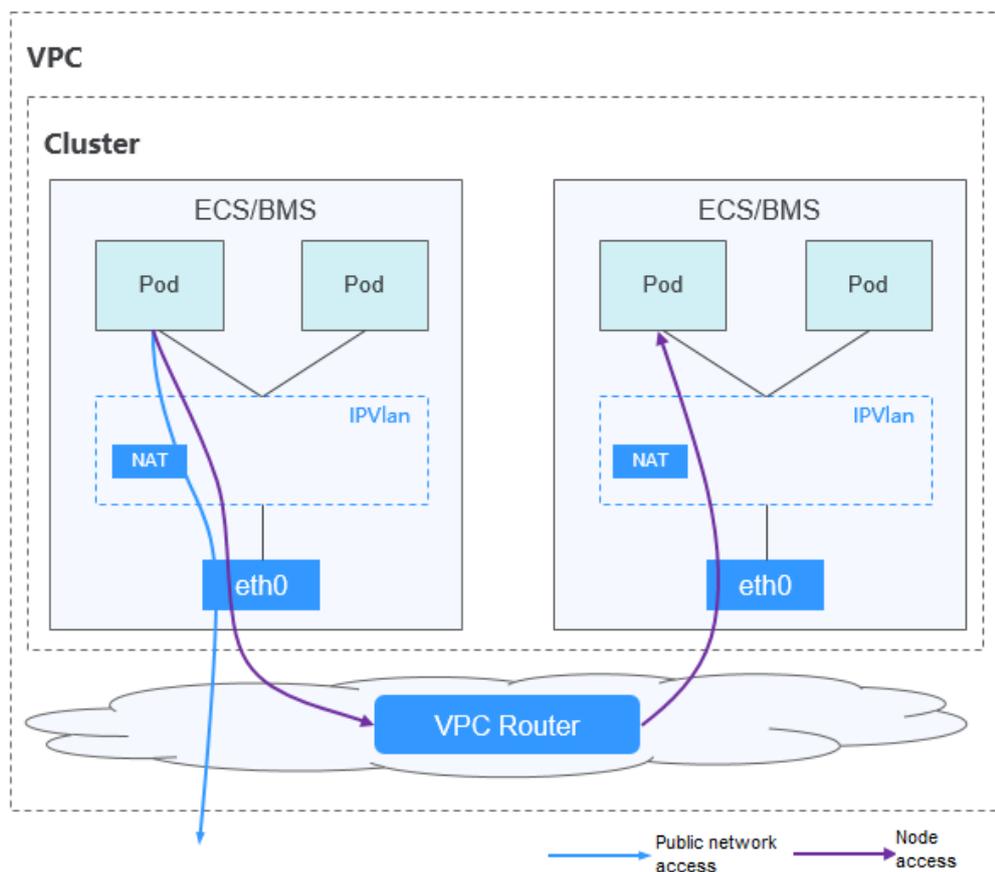
```
<p><em>Thank you for using nginx.</em></p>  
</body>  
</html>
```

## 7.2.3 Red de VPC

### Definición del modelo

La red de VPC utiliza el enrutamiento de VPC para integrarse con la red subyacente. Este modelo de red es adecuado para los escenarios de alto rendimiento. El número máximo de nodos permitidos en un clúster depende de la cuota de ruta de VPC. A cada nodo se le asigna un bloque CIDR de un tamaño fijo. Este modelo de red está libre de sobrecarga de encapsulación de túnel y supera el modelo de red de túnel contenedor. Además, como el enrutamiento de VPC incluye rutas a direcciones IP de nodo y el bloque CIDR de contenedor, se puede acceder directamente a los pods de contenedor en un clúster desde fuera del clúster.

Figura 7-7 Modelo de red de VPC



### Comunicación de pod a pod

- En el mismo nodo: Los paquetes se reenvían directamente con IPVlan.
- Entre los nodos: Los paquetes se reenvían al gateway predeterminado con las rutas predeterminadas, y luego al nodo par con las rutas de VPC.

### Ventajas y desventajas

#### Ventajas

- No se requiere encapsulación de túnel, por lo que los problemas de red son fáciles de localizar y el rendimiento es alto.
- Las redes externas de una VPC se pueden conectar directamente a las direcciones IP de contenedor.

#### Desventajas

- El número de nodos está limitado por la cuota de ruta de VPC.
- A cada nodo se le asigna un bloque CIDR de un tamaño fijo, lo que conduce a un desperdicio de direcciones IP en el bloque CIDR de contenedor.
- Los pods no pueden usar directamente funcionalidades como EIP y grupos de seguridad.

### Escenarios posibles

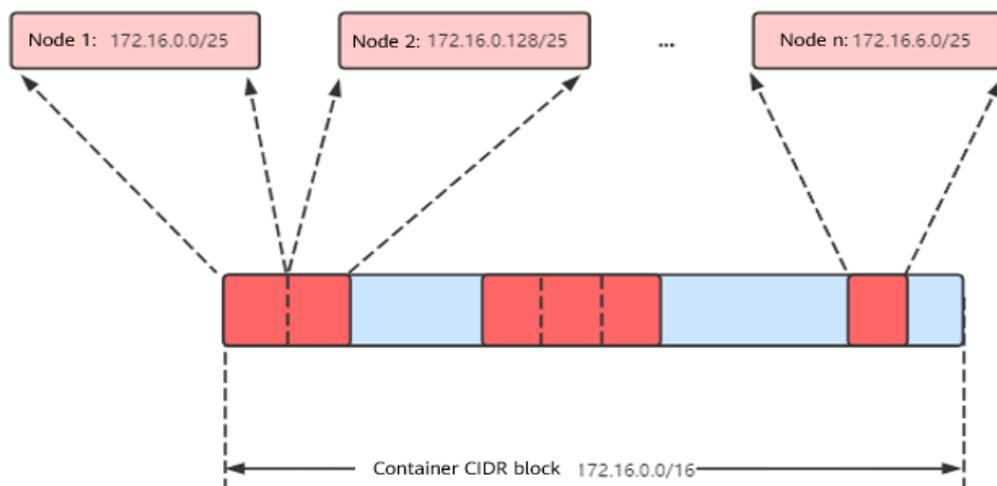
- Requisitos de alto rendimiento: como no se requiere encapsulación de túnel, el modelo de red de VPC ofrece un rendimiento cercano al de una red de VPC en comparación con el modelo de red de túnel de contenedor. Por lo tanto, el modelo de red de VPC es aplicable a escenarios que tienen altos requisitos de rendimiento, como el cómputo de IA y el cómputo de big data.
- Redes de la escala pequeña y mediana: La red de VPC está limitada por la cuota de ruta de VPC. Actualmente, se admite un máximo de 200 nodos de forma predeterminada. Si hay requisitos de red a gran escala, puede aumentar la cuota de ruta de VPC.

### Gestión de direcciones IP de contenedores

La red de VPC asigna direcciones IP de contenedor de acuerdo con las siguientes reglas:

- El bloque CIDR de contenedor se asigna por separado.
- Las direcciones IP se asignan por nodo. Un bloque CIDR con un tamaño fijo (que es configurable) se asigna a cada nodo en un grupo desde el bloque CIDR de contenedor.
- El bloque CIDR de contenedor asigna cíclicamente bloques CIDR a nuevos nodos en secuencia.
- Los pods programados para un nodo se asignan cíclicamente direcciones IP de bloques CIDR asignados al nodo.

Figura 7-8 Gestión de direcciones IP de la red VPC



Número máximo de nodos que se pueden crear en el clúster usando la red de VPC = Número de direcciones IP en el bloque CIDR de contenedor / Número de direcciones IP en el bloque CIDR asignado al nodo por el bloque CIDR de contenedor

Por ejemplo, si el bloque CIDR de contenedor es 172.16.0.0/16, el número de direcciones IP es 65536. La máscara del bloque CIDR de contenedor asignado al nodo es 25. Es decir, el número de direcciones IP de contenedor en cada nodo es 128. Por lo tanto, se puede crear un máximo de 512 (65536/128) nodos. Además, el número de nodos que se pueden crear en un clúster también depende de la red de nodos y de la escala del clúster.

## Recomendación para la planificación de bloques CIDR

Como se describe en [Estructura de red de clústeres](#), las direcciones de red de un clúster se pueden dividir en tres partes: red de nodo, red de contenedor y red de servicio. Al planificar direcciones de red, tenga en cuenta los siguientes aspectos:

- Los tres bloques CIDR no pueden superponerse. De lo contrario, se produce un conflicto.
- Asegúrese de que cada bloque CIDR tenga suficientes direcciones IP.
  - Las direcciones IP en el bloque CIDR del nodo deben coincidir con la escala del clúster. De lo contrario, no se pueden crear nodos debido a la insuficiencia de direcciones IP.
  - Las direcciones IP en el bloque CIDR de contenedor deben coincidir con la escala de servicio. De lo contrario, los pods no se pueden crear debido a la insuficiencia de direcciones IP. El número de pods que se pueden crear en cada nodo también depende de otros parámetros. Para más detalles, consulte [Número máximo de pods que se pueden crear en un nodo](#).

Suponga que un clúster contiene 200 nodos y que el modelo de red es una red VPC.

En este caso, el número de direcciones IP disponibles en la subred de nodo seleccionada debe ser mayor que 200. De lo contrario, no se pueden crear nodos debido a la insuficiencia de direcciones IP.

El bloque CIDR de contenedor es 10.0.0.0/16, y el número de direcciones IP disponibles es 65536. Como se describe en [Gestión de direcciones IP de contenedores](#), a la red de VPC se le asigna un bloque CIDR con el tamaño fijo (utilizando la máscara para determinar el número máximo de direcciones IP de contenedor asignadas a cada nodo). Por ejemplo, si el límite superior es 128, el grupo soporta un máximo de 512 (65536/128) nodos, incluidos los tres nodos maestros.

**Figura 7-9** Configuración del bloque CIDR de contenedor (al crear el clúster)

**Network Settings** Select the VPC and CIDR blocks for creating nodes and containers in the cluster.

Network Model: **VPC network** | Tunnel network | [Network Model Overview](#)  
Model used for container networking in a cluster. Not editable after creation

Number of container IP addresses reserved for each node (cannot be changed after creation):  [Learn more](#)

VPC:  [Create VPC](#)  
CIDR block used by master nodes and worker nodes in the cluster. Not editable after creation

Master Node Subnet:  [Create Subnet](#) Available Subnet IP Addresses: **250**  
Subnet used by the master node in the cluster. At least 4 IP addresses are required. Not editable after creation

Container CIDR Block: **Manually set** | Auto select | [How to plan CIDR blocks?](#)

·  ·  ·  /

Max. nodes supported by the current networking configuration: **509**

## Ejemplo del acceso a la red de VPC

Cree un clúster mediante el modelo de red de VPC. El clúster contiene un nodo.

```
$ kubectl get node
NAME          STATUS    ROLES    AGE   VERSION
192.168.0.99 Ready    <none>   9d    v1.17.17-r0-CCE21.6.1.B004-17.37.5
```

Compruebe la tabla de enrutamiento de VPC. La dirección de destino 172.16.0.0/25 es el bloque CIDR de contenedor asignado al nodo, y el salto siguiente es el nodo correspondiente. Cuando se accede a la dirección IP de contenedor, la ruta de VPC reenvía la solicitud de acceso al nodo de salto siguiente. Esto indica que el modelo de red de VPC utiliza las rutas de VPC.

**Figura 7-10** Rutas

**Routes**

[Delete](#) [Add Route](#) [Replicate Route](#) [Learn how to configure routes.](#)

<input type="checkbox"/> Destination <a href="#">?</a>	Next Hop Type <a href="#">?</a>	Next Hop <a href="#">?</a>	Type <a href="#">?</a>
Local	Local	Local	System
<input type="checkbox"/> 172.16.0.0/25	Cloud container	<a href="#">cce-ss-40633</a>	Custom

Cree una Deployment en el clúster.

```
kind: Deployment
apiVersion: apps/v1
metadata:
  name: example
  namespace: default
spec:
  replicas: 4
  selector:
    matchLabels:
      app: example
  template:
```

```

metadata:
  labels:
    app: example
spec:
  containers:
  - name: container-0
    image: 'nginx:perl'
  imagePullSecrets:
  - name: default-secret
    
```

Revisa el pod.

```

$ kubectl get pod -owide
NAME                                READY   STATUS    RESTARTS   AGE   IP
NODE                                NOMINATED NODE   READINESS GATES
example-86b9779494-l8qrw           1/1     Running   0           14s   172.16.0.6
192.168.0.99 <none>                <none>
example-86b9779494-svs8t           1/1     Running   0           14s   172.16.0.7
192.168.0.99 <none>                <none>
example-86b9779494-x8k15           1/1     Running   0           14s   172.16.0.5
192.168.0.99 <none>                <none>
example-86b9779494-zt627           1/1     Running   0           14s   172.16.0.8
192.168.0.99 <none>                <none>
    
```

En este caso, se puede acceder directamente a la dirección IP del pod desde un nodo fuera del clúster en la misma VPC. Esta es una característica de la función de la red de VPC.

También se puede acceder al pod desde un nodo en el mismo clúster o en el pod. Como se muestra en la siguiente figura, se puede acceder al pod directamente desde el contenedor.

```

$ kubectl exec -it example-86b9779494-l8qrw -- curl 172.16.0.7
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

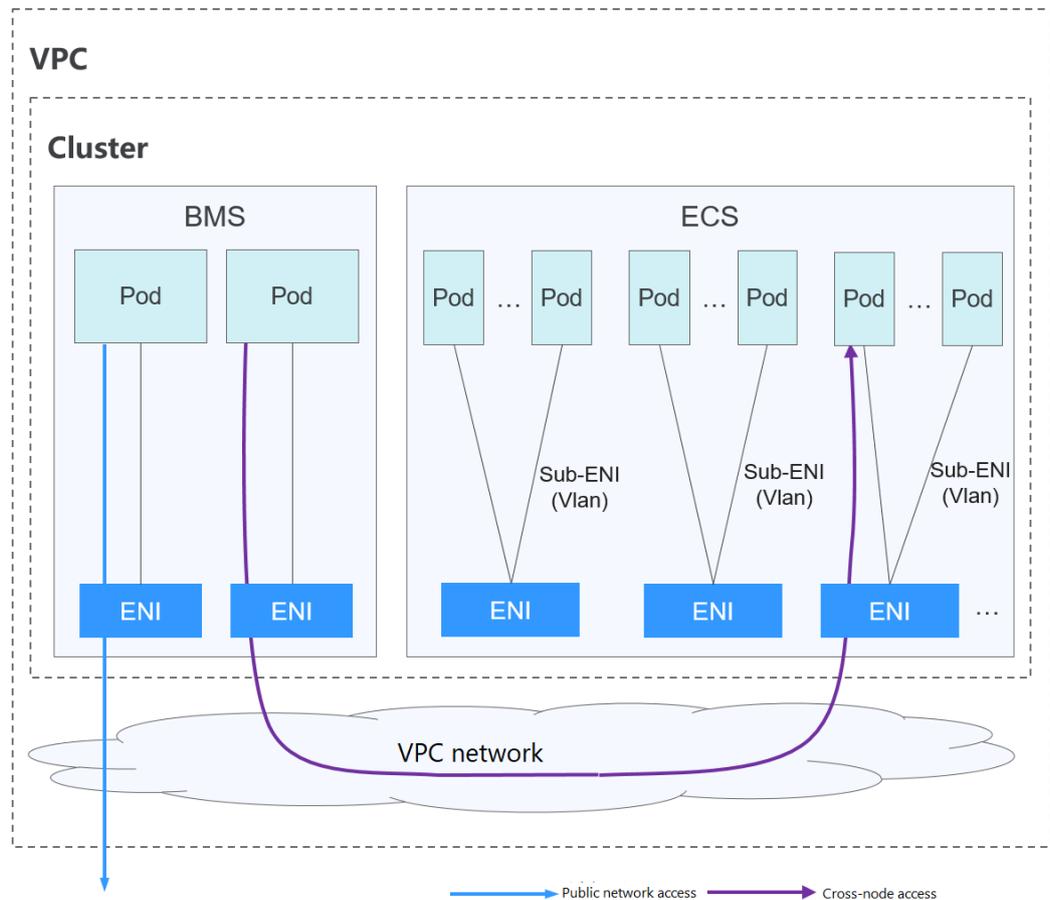
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
    
```

## 7.2.4 Cloud Native Network 2.0

### Definición del modelo

Desarrollado por CCE, Cloud Native Network 2.0 integra profundamente las interfaces de red elástica (ENI) y sub-ENI de Virtual Private Cloud (VPC). Las direcciones IP del contenedor se asignan desde el bloque CIDR de VPC. Se admite la red de paso a través de ELB para las solicitudes de acceso directo a contenedores. Los grupos de seguridad y las IP elásticas (EIP) están destinados a ofrecer un alto rendimiento.

**Figura 7-11** Cloud Native Network 2.0



### Comunicación de pod a pod

- Los pods de los nodos de BMS usan las ENI, mientras que los pods de los nodos de ECS usan las Sub-ENI. Las sub-ENI se conectan a las ENI con subinterfaces de VLAN.
- En el mismo nodo: Los paquetes se reenvían con la ENI o sub-ENI de VPC.
- A través de los nodos: Los paquetes se reenvían a través de la ENI o sub-ENI de VPC.

### Notas y restricciones

Este modelo de red solo está disponible para los clústeres de CCE Turbo.

### Ventajas y desventajas

#### Ventajas

- Como la red de contenedor utiliza directamente VPC, es fácil localizar problemas de red y proporcionar el más alto rendimiento.
- Las redes externas de una VPC se pueden conectar directamente a las direcciones IP de contenedor.
- Las capacidades de balanceo de carga, grupo de seguridad y EIP proporcionadas por VPC se pueden utilizar directamente.

#### Desventajas

La red de contenedor utiliza directamente la VPC, que ocupa el espacio de direcciones de la VPC. Por lo tanto, debe planificar correctamente el bloque CIDR de contenedor antes de crear un clúster.

## Escenarios de aplicación

- Requisitos de alto rendimiento y uso de otras capacidades de red de VPC: Cloud Native Network 2.0 utiliza directamente la VPC, que ofrece casi el mismo rendimiento que la red de VPC. Por lo tanto, es aplicable a escenarios que tienen altos requisitos de ancho de banda y de latencia, como la transmisión en vivo en línea y el seckill de comercio electrónico.
- Redes a gran escala: Cloud Native Network 2.0 admite un máximo de 2000 nodos de ECS y 100,000 contenedores.

## Gestión de direcciones IP de contenedores

En el modelo de Cloud Native Network 2.0, los nodos de BMS usan las ENI y los nodos de ECS usan las sub-ENI.

- La dirección IP del pod se asigna directamente desde la subred de VPC configurada para la red de contenedor. No es necesario asignar un segmento de red pequeño independiente al nodo.
- Para agregar un nodo de ECS a un clúster, enlaza primero la ENI que lleva la sub-ENI. Después de enlazar la ENI, puede enlazar la sub-ENI.
- Número de las ENI enlazadas a un nodo de ECS: **Número máximo de sub-ENIs que pueden enlazarse al nodo/64**. El valor se redondea hacia arriba.
- ENI enlazada a un nodo de ECS = **Número de ENIs utilizadas para soportar sub-ENIs + Número de sub-ENIs actualmente utilizadas por los pods + Número de sub-ENIs preenlazadas**
- ENIs unidas a un nodo de BMS = **Número de ENIs utilizadas actualmente por los pods + Número de ENIs preenlazadas**
- Cuando se crea un pod, se asigna aleatoriamente una ENI disponible desde el grupo de ENI de preenlace del nodo.
- Cuando se elimina el pod, la ENI se libera de nuevo al grupo de ENI del nodo.
- Cuando se elimina un nodo, las ENI se liberan de nuevo en el grupo y las sub-ENI se eliminan.

Actualmente, el modelo de Cloud Native Network 2.0 es compatible con las políticas de enlace previo ENI **dinámicas y basadas en umbrales**. En la siguiente tabla se enumeran los escenarios.

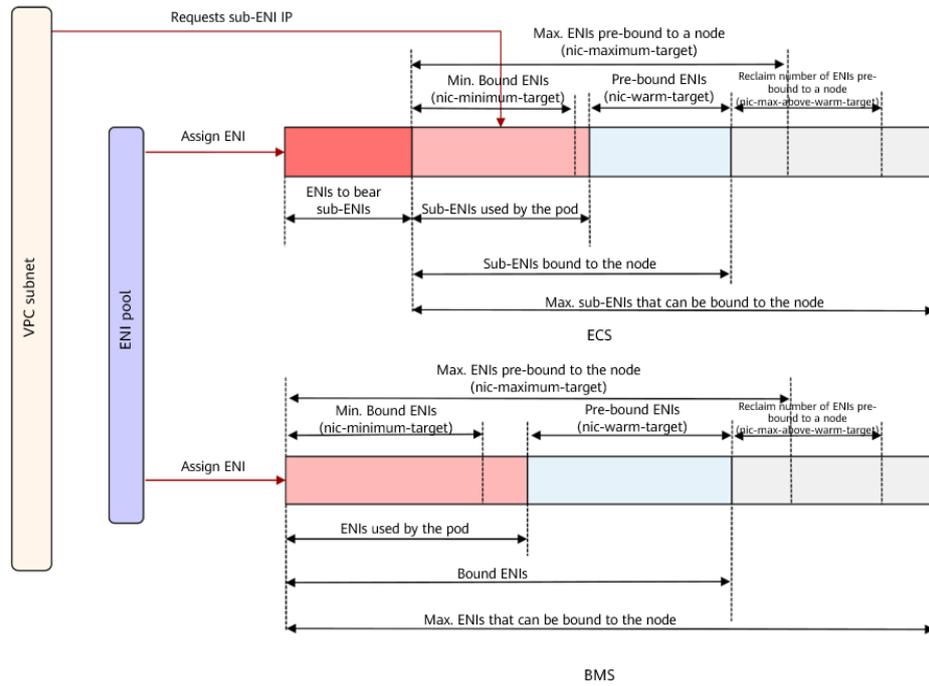
**Tabla 7-2** Comparación entre las políticas pre-vinculantes de ENI

Política	Política de pre-vinculación dinámica de ENI (predeterminada)	Política de pre-vinculación ENI basada en umbral
Política de gestión	<p><b>nic-minimum-target:</b> número mínimo de ENIs (no utilizadas + utilizadas) enlazadas a un nodo</p> <p><b>nic-maximum-target:</b> si el número de ENIs enlazadas a un nodo excede el valor de este parámetro, el sistema no vincula de forma proactiva las ENIs.</p> <p><b>Pre-bound ENIs:</b> ENIs adicionales que estarán preenlazadas a un nodo</p> <p><b>nic-max-above-warm-target:</b> Las ENI no están unidas y se recuperan solo cuando el número de ENI inactivas menos el número de <b>nic-warm-target</b> es mayor que el umbral.</p>	<p><b>Low threshold of the number of bound ENIs:</b> número mínimo de ENIs (no utilizadas + utilizadas) enlazadas a un nodo</p> <p><b>High threshold of the number of bound ENIs:</b> número máximo de ENIs que se pueden enlazar a un nodo. Si el número de ENI enlazadas a un nodo excede el valor de este parámetro, el sistema desvincula los ENI inactivas.</p>
Escenario de aplicación	<p>Acelera el inicio del pod al tiempo que mejora la utilización de los recursos IP. Este modo se aplica a escenarios en los que el número de direcciones IP en el segmento de red contenedor es insuficiente.</p> <p>Para obtener más información sobre los parámetros anteriores, consulte <a href="#">ENIs de previnculación para clústeres de CCE Turbo</a>.</p>	<p>Se aplica a escenarios en los que el número de direcciones IP en el bloque CIDR contenedor es suficiente y el número de pods en los nodos cambia bruscamente pero se fija en un cierto rango.</p>

 **NOTA**

- Para clústeres de 1.19.16-r2, 1.21.5-r0, 1.23.3-r0 a 1.19.16-r4, 1.21.7-r0 y 1.23.5-r0, solo se admiten los parámetros **nic-minimum-target** y **nic-warm-target**. La política de previnculación basada en umbrales tiene prioridad sobre la política de previnculación de ENI dinámica.
- Para clústeres de 1.19.16-r4, 1.21.7-r0, 1.23.5-r0, 1.25.1-r0 y posteriores, se admiten los cuatro parámetros anteriores. La política de pre-vinculación de ENI dinámica tiene prioridad sobre la política de pre-vinculación basada en umbrales.

**Figura 7-12** Política de previnculación de ENI dinámica



CCE proporciona cuatro parámetros para la política dinámica de pre-encuadración de ENI. Establezca estos parámetros correctamente.

**Tabla 7-3** Parámetros de la política de pre-encuadración de ENI dinámica

Parámetro	Valor predeterminado	Descripción	Sugerencia
nic-minimum-target	10	<p>Número mínimo de ENI enlazadas a un nodo. El valor puede ser un número o un porcentaje.</p> <ul style="list-style-type: none"> <li>● Valor: El valor debe ser un entero positivo. Por ejemplo, 10 indica que al menos 10 ENI están unidas a un nodo. Si se excede la cuota de ENI de un nodo, se utiliza la cuota de ENI.</li> <li>● Porcentaje: El valor oscila entre el 1% y el 100%. Por ejemplo, 10%. Si la cuota de ENI de un nodo es 128, al menos 12 ENI (redondeados hacia abajo) están unidas al nodo.</li> </ul> <p>Establezca tanto <b>nic-minimum-target</b> como <b>nic-maximum-target</b> en el mismo valor o porcentaje.</p>	Establezca estos parámetros en función del número de pods.
nic-maximum-target	0	<p>Si el número de ENI enlazadas a un nodo excede el valor de <b>nic-maximum-target</b>, el sistema no enlaza de forma proactiva las ENI.</p> <p>Si el valor de este parámetro es mayor o igual que el valor de <b>nic-minimum-target</b>, se activa la comprobación del número máximo de ENIs preenlazadas. De lo contrario, la comprobación está deshabilitada. El valor puede ser un número o un porcentaje.</p> <ul style="list-style-type: none"> <li>● Valor: El valor debe ser un entero positivo. Por ejemplo, 0. La comprobación del número máximo de ENIs preenlazadas está desactivada. Si se excede la cuota de ENI de un nodo, se utiliza la cuota de ENI.</li> <li>● Porcentaje: El valor oscila entre el 1% y el 100%. Por ejemplo, 50%. Si la cuota de ENI de un nodo es 128, el número máximo de ENIs preenlazadas es 64 (redondeado hacia abajo).</li> </ul> <p>Establezca tanto <b>nic-minimum-target</b> como <b>nic-maximum-target</b> en el mismo valor o porcentaje.</p>	Establezca estos parámetros en función del número de pods.

Parámetro	Valor predeterminado	Descripción	Sugerencia
nic-warm-target	2	<p>Las ENIs adicionales estarán pre-enlazadas después de que el <b>nic-minimum-target</b> se haya usado en un pod. El valor solo puede ser un número.</p> <p>Cuando el valor de <b>nic-warm-target</b> + el número de ENIs enlazadas es mayor que el valor de <b>nic-maximum-target</b>, el sistema pre-enlazará ENIs en función de la diferencia entre el valor de <b>nic-maximum-target</b> y el número de ENIs enlazadas.</p>	<p>Establezca este parámetro en el número de pods que se pueden escalar instantáneamente en 10 segundos.</p>
nic-max-above-warm-target	2	<p>Solo cuando el número de ENIs inactivas en un nodo menos el valor de <b>nic-warm-target</b> es mayor que el umbral, las ENIs preenlazadas no se enlazarán y se reclamarán. El valor solo puede ser un número.</p> <ul style="list-style-type: none"> <li>● Establecer un valor mayor de este parámetro ralentiza el reciclaje de las ENI inactivas y acelera el inicio de pod. Sin embargo, el uso de direcciones IP disminuye, especialmente cuando las direcciones IP son insuficientes. Por lo tanto, <b>tenga cuidado al aumentar el valor de este parámetro.</b></li> <li>● Establecer un valor más pequeño de este parámetro acelera el reciclaje de las ENI inactivas y mejora el uso de la dirección IP. Sin embargo, cuando un gran número de pods aumenta instantáneamente, el inicio de algunos pods se ralentiza.</li> </ul>	<p>Establezca este parámetro en función de la diferencia entre el número de pods que se escalan con frecuencia en la mayoría de los nodos en cuestión de minutos y el número de pods que se escalan instantáneamente en la mayoría de los nodos en 10 segundos.</p>

 **NOTA**

Los parámetros anteriores admiten la configuración global en el nivel del clúster y la configuración diferenciada en el nivel del grupo de nodos. Este último tiene prioridad sobre el primero.

El componente de red de contenedor mantiene un grupo de ENI preenlazadas escalable para cada nodo. El componente comprueba y calcula el número de ENI preenlazadas o ENI inactivas cada 10 segundos.

- **Número de ENIs preenlazadas = min(nic-máximo-objetivo - Número de ENIs vinculadas, máx(nic-mínimo-objetivo - Número de ENIs vinculadas, nic-warm-target - Número de ENIs inactivas)**

- **Número de ENIs a no enlazar** =  $\min(\text{Número de ENIs inactivas} - \text{nic-warm-target-nic-max-above-warm-target}, \text{número de ENIs unidas} - \text{nic-minimum-target})$

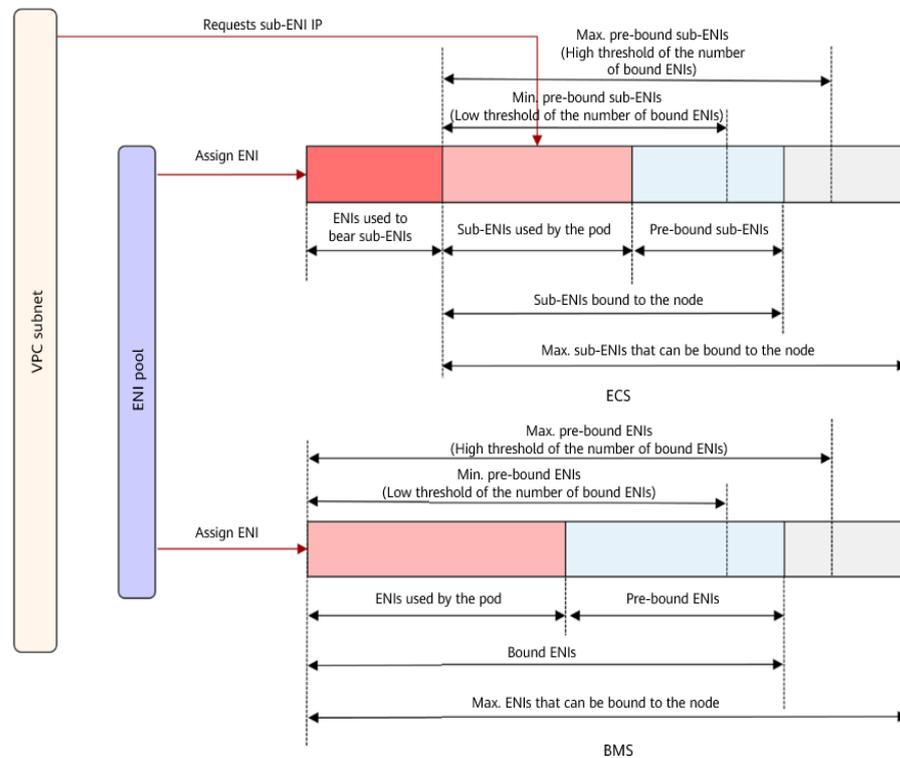
El número de ENIs pre-enlazadas en el nodo permanece en el rango siguiente:

- **Número mínimo de ENIs a ser pre-enlazadas** =  $\min(\text{máx}(\text{nic-minimum-target} - \text{número de ENIs enlazadas}, \text{nic-warm-target}), \text{nic-maximum-target} - \text{número de ENIs enlazadas})$
- **Número máximo de ENIs a ser pre-enlazadas** =  $\text{máx}(\text{nic-warm-target} + \text{nic-max-above-warm-target}, \text{número de ENIs enlazadas} - \text{nic-minimum-target})$

Cuando se crea un pod, una ENI inactiva (la más antigua no utilizada) se asigna preferentemente desde el grupo. Si no hay ENI inactiva disponible, una nueva sub-ENI está enlazada al pod.

Cuando se elimina el pod, la ENI correspondiente se libera de nuevo al grupo de ENI pre-enlazadas del nodo, entra en un periodo de enfriamiento de 2 minutos, y se puede vincular a otro pod. Si la ENI no está vinculada a ningún pod en 2 minutos, se liberará.

**Figura 7-13** Política basada en umbrales



CCE proporciona un parámetro de configuración para los algoritmos de umbral. Puede establecer este parámetro en función del plan de servicio, la escala del clúster y el número de ENI que se pueden enlazar a un nodo.

- **Low threshold of the number of bound ENIs:** El valor predeterminado es **0** que indica el número mínimo de ENIs (no utilizadas + utilizadas) enlazadas a un nodo. Número mínimo de ENIs pre-enlazadas en un nodo de ECS = Número de ENIs enlazadas al nodo en el umbral bajo x Número de subENIs en el nodo. Número mínimo de ENIs pre-enlazadas en un nodo de BMS = Número de ENIs enlazadas al nodo en el umbral bajo x Número de ENIs en el nodo.

- **High threshold of the number of bound ENIs:** El valor predeterminado es **0** que indica el número máximo de ENI que se pueden enlazar a un nodo. Si el número de ENI enlazadas a un nodo excede el valor de este parámetro, el sistema desvincula los ENI inactivos. Número máximo de ENIs preenlazadas en un nodo de ECS = Número de ENIs enlazadas en el umbral alto x Número de subENIs en el nodo. Número máximo de ENIs preenlazadas en un nodo de BMS = Número de ENIs enlazadas en el umbral alto x Número de ENIs en el nodo.

El componente de red de contenedor mantiene un grupo de ENI escalable para cada nodo.

- Si el número de ENIs enlazadas (ENIs usadas + ENIs preenlazadas) es menor que el número de ENIs preenlazadas en el umbral bajo, las ENIs son enlazadas hasta que los dos números son iguales.
- Si el número de ENIs enlazadas (ENIs usadas + ENIs preenlazadas) es mayor que el número de ENIs preenlazadas en el umbral alto y el número de ENIs preenlazadas es mayor que 0, las ENIs preenlazadas que no se utilicen durante más de 2 minutos se liberarán periódicamente hasta que el número de ENIs enlazadas = Número de ENIs preenlazadas en el umbral alto o el número de ENIs usadas sea mayor que el número de ENIs preenlazadas en el umbral alto y el número de ENIs preenlazadas en el nodo es 0.

## Recomendación para la planificación de bloques CIDR

Como se describe en [Estructura de red de clústeres](#), las direcciones de red de un clúster se pueden dividir en tres partes: red de nodo, red de contenedor y red de servicio. Al planificar direcciones de red, tenga en cuenta los siguientes aspectos:

- Los tres bloques CIDR no pueden superponerse. De lo contrario, se produce un conflicto. Todas las subredes (incluidas las creadas a partir del bloque CIDR secundario) en la VPC donde reside el clúster no pueden entrar en conflicto con los bloques CIDR de contenedor y de Service.
- Asegúrese de que cada bloque CIDR tenga suficientes direcciones IP.
  - Las direcciones IP en el bloque CIDR del nodo deben coincidir con la escala del clúster. De lo contrario, no se pueden crear nodos debido a la insuficiencia de direcciones IP.
  - Las direcciones IP en el bloque CIDR de contenedor deben coincidir con la escala de servicio. De lo contrario, los pods no se pueden crear debido a la insuficiencia de direcciones IP.

En el modelo de Cloud Native Network 2.0, el bloque CIDR de contenedor y el bloque CIDR del nodo comparten las direcciones de red en una VPC. Se recomienda que la subred de contenedor y la subred de nodo no utilicen la misma subred. De lo contrario, es posible que no se creen contenedores o nodos debido a la insuficiencia de recursos de la IP.

Además, se puede agregar una subred al bloque CIDR de contenedor después de crear un clúster para aumentar el número de direcciones IP disponibles. En este caso, asegúrese de que la subred agregada no entre en conflicto con otras subredes en el bloque CIDR de contenedor.

**Figura 7-14** Configuración de los bloques CIDR (al crear el clúster)

**Network Settings** Select the VPC and CIDR blocks for creating nodes and containers in the cluster.

Network Model: **Cloud Native Network 2.0** ?  
Model used for container networking in a cluster. **Not editable after creation**

VPC: **vpc-100390292 (10.121.0.0/16)** C Create VPC  
CIDR block used by master nodes and worker nodes in the cluster. **Not editable after creation**

Master Node Subnet: **subnet-2467 (10.121.2.0/24)** C Create Subnet Available Subnet IP Addresses: **246**  
Subnet used by the master node in the cluster. At least 4 IP addresses are required. **Not editable after creation**

Pod Subnet: **subnet-2467 (10.121.2.0/24)** C Create Subnet Available IP addresses for pods: **246**  
The subnet you selected determines the maximum number of containers in the cluster. After the cluster is created, you can add a subnet.

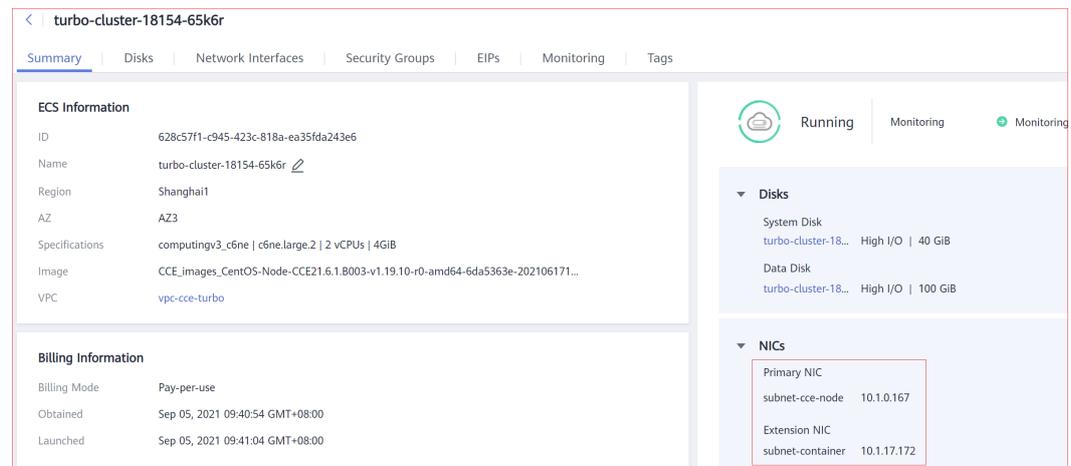
Service CIDR: **10** . **247** . **0** . **0** / **16** Max. Services allowed by this CIDR block: **65,536**  
CIDR block for Services used by containers in the cluster to access each other, which determines the maximum number of Services. **Not editable after creation**

## Ejemplo de acceso a Cloud Native Network 2.0

Cree un clúster de CCE Turbo, que contenga tres nodos de ECS.

Acceda a la página de detalles de un nodo. Puede ver que el nodo tiene una ENI principal y una ENI extendida, y ambas son las ENI. La ENI extendida pertenece al bloque CIDR de contenedor y se utiliza para montar un sub-ENI en el pod.

**Figura 7-15** ENIs de nodo



Cree una Deployment en el clúster.

```
kind: Deployment
apiVersion: apps/v1
metadata:
  name: example
  namespace: default
spec:
  replicas: 6
  selector:
    matchLabels:
      app: example
  template:
    metadata:
      labels:
        app: example
    spec:
      containers:
```

```

- name: container-0
  image: 'nginx:perl'
  resources:
    limits:
      cpu: 250m
      memory: 512Mi
    requests:
      cpu: 250m
      memory: 512Mi
  imagePullSecrets:
    - name: default-secret
    
```

Vea el pod creado.

```

$ kubectl get pod -owide
NAME                                READY   STATUS    RESTARTS   AGE   IP
NODE                                NOMINATED NODE   READINESS GATES
example-5bdc5699b7-54v7g           1/1     Running   0           7s    10.1.18.2
10.1.0.167                         <none>          <none>
example-5bdc5699b7-6dzz5           1/1     Running   0           7s    10.1.18.216
10.1.0.186                         <none>          <none>
example-5bdc5699b7-gq7xs           1/1     Running   0           7s    10.1.16.63
10.1.0.144                         <none>          <none>
example-5bdc5699b7-h9rvb           1/1     Running   0           7s    10.1.16.125
10.1.0.167                         <none>          <none>
example-5bdc5699b7-s9fts           1/1     Running   0           7s    10.1.16.89
10.1.0.144                         <none>          <none>
example-5bdc5699b7-swq6q           1/1     Running   0           7s    10.1.17.111
10.1.0.167                         <none>          <none>
    
```

Las direcciones IP de todos los pods son sub-ENIs, que se montan en la ENI (ENI extendida) del nodo.

Por ejemplo, la ENI extendida del nodo 10.1.0.167 es 10.1.17.172. En la página **Network Interfaces** de la consola de red, puede ver que tres sub-ENI están montadas en la ENI extendida 10.1.17.172, que es la dirección IP del pod.

**Figura 7-16** ENIs de pod



En la VPC, se puede acceder con éxito a la dirección IP del pod.

## 7.3 Service

### 7.3.1 Descripción general

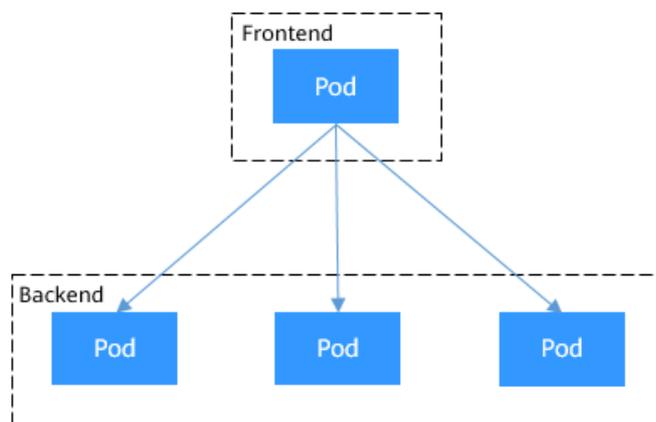
#### Acceso directo a un pod

Después de crear un pod, pueden producirse los siguientes problemas si accede directamente al pod:

- El pod puede ser borrado y recreado en cualquier momento por un controlador tal como un Deployment, y el resultado de acceder al pod se vuelve impredecible.
- La dirección IP del pod se asigna solo después de iniciar el pod. Antes de iniciar el pod, se desconoce la dirección IP del pod.
- Una aplicación suele estar compuesta por varios pods que ejecutan la misma imagen. Acceder a los pods uno por uno no es eficiente.

Por ejemplo, una aplicación utiliza Deployments para crear el frontend y el backend. El frontend llama al backend para el cómputo, como se muestra en **Figura 7-17**. Tres pods están funcionando en el backend, que son independientes y reemplazables. Cuando se vuelve a crear un pod de backend, se asigna al nuevo pod una nueva dirección IP, de la cual el pod de frontend no está al tanto.

**Figura 7-17** Acceso dentro de pod

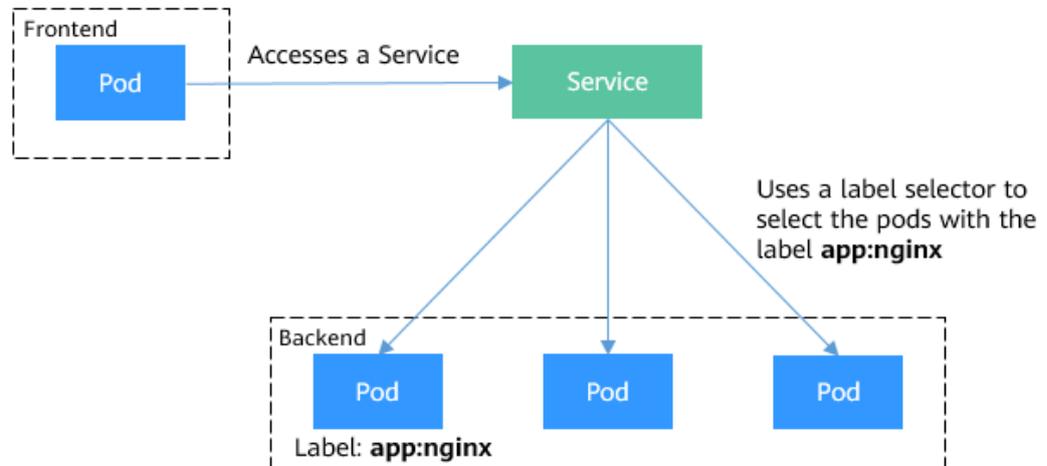


## Uso de Services para el acceso a pods

Los Services de Kubernetes se utilizan para resolver los problemas de acceso a pods anteriores. Un Service tiene una dirección IP fija. (Cuando se crea un clúster de CCE, se establece un bloque CIDR de Service, que se utiliza para asignar direcciones IP a Services.) Un Service reenvía las solicitudes de acceso al Service a los pods basados en etiquetas y, al mismo tiempo, realiza el equilibrio de carga para estos pods.

En el ejemplo anterior, se agrega un Service para que el pod frontend acceda a los pods de backend. De esta manera, el pod frontend no necesita estar al tanto de los cambios en los pods backend, como se muestra en **Figura 7-18**.

**Figura 7-18** Acceso a pods con un Service



## Tipos de servicio

Kubernetes le permite especificar un servicio de un tipo requerido. Los valores y acciones de los diferentes tipos de Servicios son los siguientes:

- **ClusterIP**

Un Service de ClusterIP permite que las cargas de trabajo del mismo clúster utilicen sus nombres de dominio internos del clúster para tener acceso entre sí.

- **NodePort**

Un Service de NodePort está expuesto en la IP de cada nodo en un puerto estático. Un Service de ClusterIP, al que se enruta el Service de NodePort, se crea automáticamente. Al solicitar `<NodeIP>:<NodePort>`, puede acceder a un Service de NodePort desde fuera del clúster.

- **LoadBalancer**

Los LoadBalancer Service pueden acceder a cargas de trabajo desde la red pública con ELB, que es más fiable que el acceso basado en EIP.

- **DNAT**

Un gateway de DNAT traduce direcciones para los nodos de clúster y permite que varios nodos de clúster compartan una EIP. Los servicios de DNAT ofrecen una mayor confiabilidad que los servicios de NodePort basados en EIP. No es necesario vincular una EIP a un solo nodo y las solicitudes se pueden distribuir a la carga de trabajo incluso cualquiera de los nodos internos está inactivo.

## externalTrafficPolicy (afinidad del Service)

Para un Service de NodePort y de LoadBalancer las solicitudes se envían primero al puerto del nodo, luego al Service y, finalmente, al pod que respalda el Service. El pod de respaldo puede no estar ubicado en el nodo que recibe las solicitudes. De forma predeterminada, se puede acceder a la carga de trabajo de backend desde cualquier dirección IP de nodo y puerto de servicio. Si el pod no está en el nodo que recibe la solicitud, la solicitud se redirigirá al nodo donde se encuentra el pod, lo que puede causar pérdida de rendimiento.

**externalTrafficPolicy** es un parámetro de configuración del Service.

```
apiVersion: v1
kind: Service
```

```

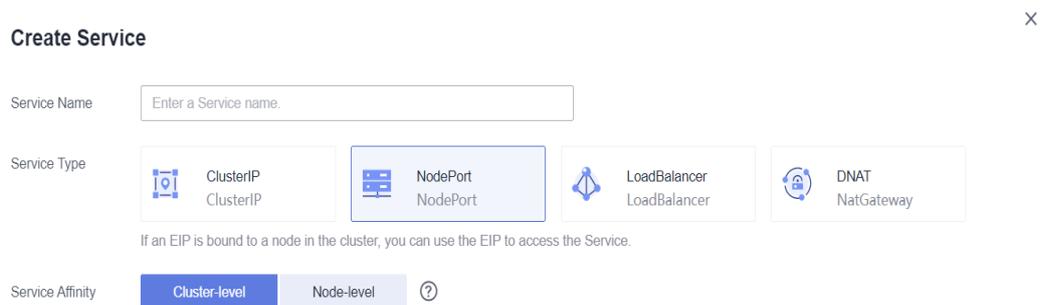
metadata:
  name: nginx-nodeport
spec:
  externalTrafficPolicy: local
  ports:
  - name: service
    nodePort: 30000
    port: 80
    protocol: TCP
    targetPort: 80
  selector:
    app: nginx
  type: NodePort
    
```

Si el valor de **externalTrafficPolicy** es **local**, las solicitudes enviadas desde *Node IP address:Service port* se reenviarán solo al pod en el nodo local. Si el nodo no tiene un pod, las solicitudes se suspenden.

Si el valor de **externalTrafficPolicy** es **cluster**, las solicitudes se reenvían dentro del clúster y se puede acceder a la carga de trabajo de backend desde cualquier dirección IP de nodo y puerto de servicio.

Si **externalTrafficPolicy** no está definido, se utiliza el valor predeterminado **cluster**.

Puede establecer este parámetro al crear un Service del tipo NodePort en la consola de CCE.



Los valores de **externalTrafficPolicy** son los siguientes:

- **cluster**: Las direcciones IP y los puertos de acceso de todos los nodos de un clúster pueden acceder a la carga de trabajo asociada con el Service. El acceso al Service causará una pérdida de rendimiento debido a la redirección de la ruta y no se puede obtener la dirección IP de origen del cliente.
- **local**: Solo la dirección IP y el puerto de acceso del nodo donde se encuentra la carga de trabajo pueden acceder a la carga de trabajo asociada con el Service. El acceso al Service no causará pérdida de rendimiento debido a la redirección de la ruta, y se puede obtener la dirección IP de origen del cliente.

## 7.3.2 ClusterIP

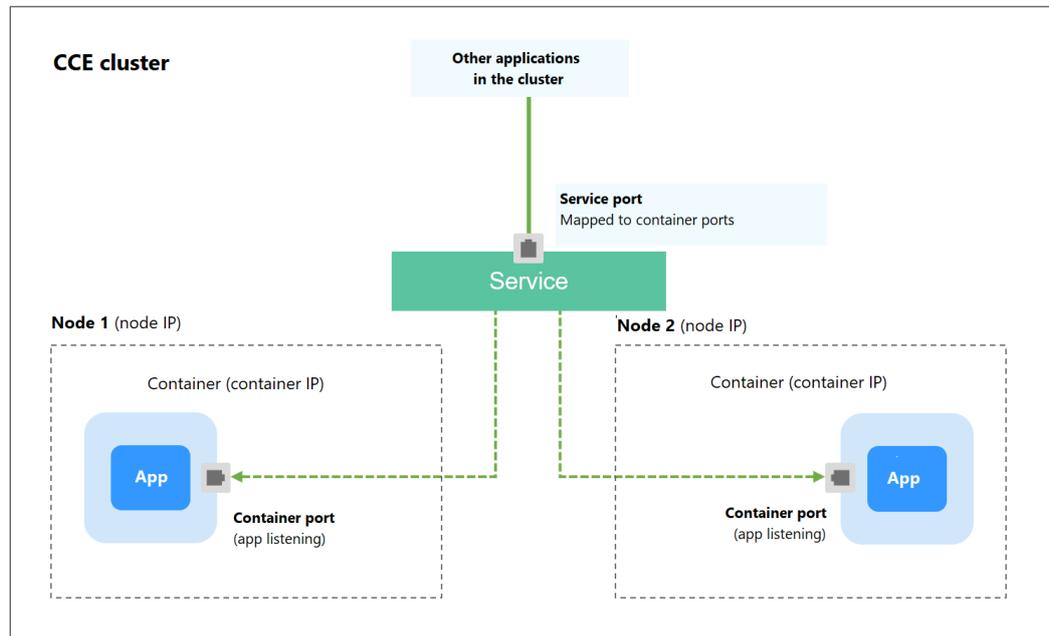
### Escenario

Los Services de ClusterIP permiten que las cargas de trabajo del mismo clúster utilicen sus nombres de dominio internos de clúster para tener acceso entre sí.

El formato de nombre de dominio interno de clúster es `<Service name>.<Namespace of the workload>.svc.cluster.local:<Port>`. Por ejemplo, `nginx.default.svc.cluster.local:80`.

**Figura 7-19** muestra las relaciones de mapeo entre canales de acceso, puertos de contenedor y puertos de acceso.

Figura 7-19 Acceso dentro del clúster (ClusterIP)



## Creación de un Service de ClusterIP

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** Elija **Networking** en el panel de navegación y haga clic en **Create Service** en la esquina superior derecha.

**Paso 3** Establezca los parámetros de acceso dentro del clúster.

- **Service Name:** Nombre del Service, que puede ser el mismo que el nombre de la carga de trabajo.
- **Service Type:** Seleccione **ClusterIP**.
- **Namespace:** Espacio de nombres al que pertenece la carga de trabajo.
- **Selector:** Agregue una etiqueta y haga clic en **Add**. Un Service selecciona un pod basado en la etiqueta agregada. También puede hacer clic en **Reference Workload Label** para hacer referencia a la etiqueta de una carga de trabajo existente. En el cuadro de diálogo que se muestra, seleccione una carga de trabajo y haga clic en **OK**.
- **IPv6:** Esta función está deshabilitada por defecto. Una vez habilitada esta función, la dirección IP del clúster del Service cambia a una dirección IPv6. Para obtener más información, consulte [¿Cómo creo un clúster de doble pila IPv4/IPv6? Este parámetro solo está disponible en clústeres de v1.15 o posterior con IPv6 habilitado \(establecido durante la creación del clúster\).](#)
- **Configuraciones del puerto**
  - **Protocol:** protocolo utilizado por el Service.
  - **Service Port:** puerto utilizado por el Service. El número de puerto se encuentra dentro del rango de 1 a 65535.
  - **Container Port:** puerto en el que escucha la carga de trabajo. Por ejemplo, Nginx utiliza el puerto 80 de forma predeterminada.

**Paso 4** Haga clic en **OK**.

---Fin

## Configuración del tipo de acceso con kubectl

Puede ejecutar comandos de kubectl para establecer el tipo de acceso (Service). Esta sección utiliza una carga de trabajo de Nginx como ejemplo para describir cómo implementar el acceso intracluster usando kubectl.

**Paso 1** Utilice kubectl para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).

**Paso 2** Cree y edite los archivos **nginx-deployment.yaml** y **nginx-clusterip-svc.yaml**.

Los nombres de archivo están definidos por el usuario. **nginx-deployment.yaml** y **nginx-clusterip-svc.yaml** son simplemente nombres de archivo de ejemplo.

### vi nginx-deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx:latest
        name: nginx
        imagePullSecrets:
        - name: default-secret
```

### vi nginx-clusterip-svc.yaml

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app: nginx
  name: nginx-clusterip
spec:
  ports:
  - name: service0
    port: 8080 # Port for accessing a Service.
    protocol: TCP # Protocol used for accessing a Service. The value
    can be TCP or UDP.
    targetPort: 80 # Port used by a Service to access the target
    container. This port is closely related to the applications running in a
    container. In this example, the Nginx image uses port 80 by default.
  selector: # Label selector. A Service selects a pod based on
  the label and forwards the requests for accessing the Service to the pod. In this
  example, select the pod with the app:nginx label.
    app: nginx
  type: ClusterIP # Type of a Service. ClusterIP indicates that a
  Service is only reachable from within the cluster.
```

**Paso 3** Cree una carga de trabajo.

**kubectl create -f nginx-deployment.yaml**

Si se muestra información similar a la siguiente, se ha creado la carga de trabajo.

```
deployment "nginx" created
```

### kubectl get po

Si se muestra la información similar a la siguiente, la carga de trabajo se está ejecutando.

NAME	READY	STATUS	RESTARTS	AGE
nginx-2601814895-znhbr	1/1	Running	0	15s

## Paso 4 Cree un Service.

### kubectl create -f nginx-clusterip-svc.yaml

Si se muestra información similar a la siguiente, se está creando el Service.

```
service "nginx-clusterip" created
```

### kubectl get svc

Si se muestra información similar a la siguiente, se ha creado el Service y se ha asignado una dirección IP interna del clúster al Service.

```
# kubectl get svc
NAME                TYPE          CLUSTER-IP    EXTERNAL-IP  PORT(S)    AGE
kubernetes          ClusterIP     10.247.0.1    <none>       443/TCP    4d6h
nginx-clusterip     ClusterIP     10.247.74.52 <none>       8080/TCP   14m
```

## Paso 5 Acceda a un Service.

Se puede acceder a un Service desde contenedores o nodos de un clúster.

Cree un pod, acceda al pod y ejecute el comando **curl** para acceder a *IP address:Port* o al nombre de dominio del Service, como se muestra en la siguiente figura.

El sufijo del nombre de dominio se puede omitir. En el mismo espacio de nombres, puede usar directamente **nginx-clusterip:8080** para el acceso. En otros espacios de nombres, puede usar **nginx-clusterip.default:8080** para tener acceso.

```
# kubectl run -i --tty --image nginx:alpine test --rm /bin/sh
If you do not see a command prompt, try pressing Enter.
/ # curl 10.247.74.52:8080
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

```

/ # curl nginx-clusterip.default.svc.cluster.local:8080
...
<h1>Welcome to nginx!</h1>
...
/ # curl nginx-clusterip.default:8080
...
<h1>Welcome to nginx!</h1>
...
/ # curl nginx-clusterip:8080
...
<h1>Welcome to nginx!</h1>
...
    
```

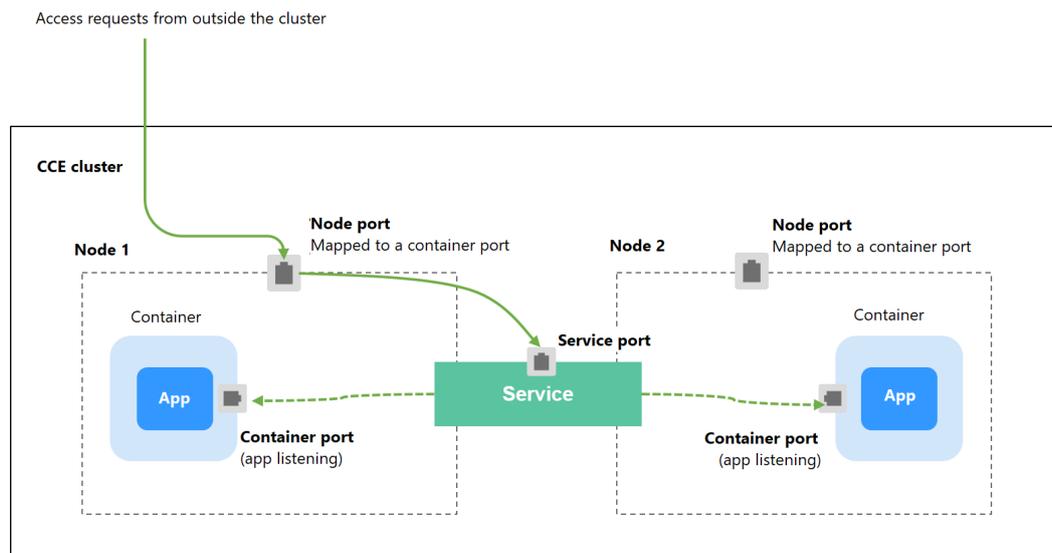
---Fin

### 7.3.3 NodePort

#### Escenario

Un Service está expuesto en la dirección IP de cada nodo en un puerto estático (NodePort). Se crea automáticamente un Service de ClusterIP, al que se enrutará el Service de NodePort. Al solicitar <NodeIP>:<NodePort>, puede acceder a un Service de NodePort desde fuera del clúster.

Figura 7-20 Acceso a NodePort



#### Notas y restricciones

- De forma predeterminada, se accede a un NodePort Service dentro de una VPC. Si necesita usar un EIP para acceder a un NodePort Service a través de redes públicas, vincule un EIP al nodo del clúster de antemano.
- Después de crear un Service, si la configuración de afinidad se cambia del nivel de clúster al nivel de nodo, la tabla de seguimiento de conexiones no se borrará. Se recomienda no modificar la configuración de afinidad del Service después de que se haya creado el Servicio. Si necesita modificarlo, vuelva a crear un Service.
- En el modo de red de VPC, cuando el contenedor A se publica con un servicio de NodePort y la afinidad del servicio se establece en el nivel de nodo (es decir, **externalTrafficPolicy** se establece en **local**), el contenedor B desplegado en el mismo

nodo no puede acceder al contenedor A con la dirección IP del nodo y el servicio de NodePort.

- Cuando se crea un servicio de NodePort en un clúster de v1.21.7 o posterior, el puerto del nodo no se muestra mediante **netstat** de forma predeterminada. Si el modo de reenvío del clúster es **iptables**, ejecute el comando **iptables -t nat -L** para ver el puerto. Si el modo de reenvío del clúster es **ipvs**, ejecute el comando **ipvsadm -nL** para ver el puerto.

## Creación de un Service de NodePort

**Paso 1** Inicie sesión en la consola de CCE y haga clic en el nombre del clúster para acceder al clúster.

**Paso 2** Elija **Networking** en el panel de navegación y haga clic en **Create Service** en la esquina superior derecha.

**Paso 3** Establezca los parámetros de acceso dentro del clúster.

- **Service Name:** Especifique un nombre de Service, que puede ser el mismo que el nombre de la carga de trabajo.
- **Service Type:** Seleccione **NodePort**.
- **Namespace:** Espacio de nombres al que pertenece la carga de trabajo.
- **Service Affinity:** Para más información, véase [externalTrafficPolicy \(afinidad del Service\)](#).
  - **Cluster level:** las direcciones IP y los puertos de acceso de todos los nodos de un clúster pueden acceder a la carga de trabajo asociada con el Service. El acceso al Service causará una pérdida de rendimiento debido a la redirección de la ruta y no se puede obtener la dirección IP de origen del cliente.
  - **Node level:** Solo la dirección IP y el puerto de acceso del nodo donde se encuentra la carga de trabajo pueden acceder a la carga de trabajo asociada con el Service. El acceso al Service no causará pérdida de rendimiento debido a la redirección de la ruta, y se puede obtener la dirección IP de origen del cliente.
- **Selector:** Agregue una etiqueta y haga clic en **Add**. Un Service selecciona un pod basado en la etiqueta agregada. También puede hacer clic en **Reference Workload Label** para hacer referencia a la etiqueta de una carga de trabajo existente. En el cuadro de diálogo que se muestra, seleccione una carga de trabajo y haga clic en **OK**.
- **IPv6:** Esta función está deshabilitada por defecto. Una vez habilitada esta función, la dirección IP del clúster del Service cambia a una dirección IPv6. Para obtener más información, consulte [¿Cómo creo un clúster de doble pila IPv4/IPv6? Este parámetro solo está disponible en clústeres de v1.15 o posterior con IPv6 habilitado \(establecido durante la creación del clúster\)](#).
- **Configuraciones del puerto**
  - **Protocol:** protocolo utilizado por el Service.
  - **Service Port:** puerto utilizado por el Service. El número de puerto se encuentra dentro del rango de 1 a 65535.
  - **Container Port:** puerto en el que escucha la carga de trabajo. Por ejemplo, Nginx utiliza el puerto 80 de forma predeterminada.
  - **Node Port:** Se recomienda seleccionar **Auto**. También puede especificar un puerto. El puerto predeterminado oscila entre 30000 y 32767.

**Paso 4** Haga clic en **OK**.

---Fin

## Uso de kubectl

Puede ejecutar comandos kubectl para establecer el tipo de acceso. En esta sección se utiliza una carga de trabajo de Nginx como ejemplo para describir cómo establecer un Service de NodePort mediante kubectl.

**Paso 1** Utilice kubectl para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).

**Paso 2** Cree y edite los archivos **nginx-deployment.yaml** y **nginx-nodeport-svc.yaml**.

Los nombres de archivo están definidos por el usuario. **nginx-deployment.yaml** y **nginx-nodeport-svc.yaml** son simplemente nombres de archivo de ejemplo.

### vi nginx-deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx:latest
        name: nginx
        imagePullSecrets:
        - name: default-secret
```

### vi nginx-nodeport-svc.yaml

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app: nginx
  name: nginx-nodeport
spec:
  ports:
  - name: service
    nodePort: 30000 # Node port. The value ranges from 30000 to 32767.
    port: 8080 # Port for accessing a Service.
    protocol: TCP # Protocol used for accessing a Service. The value can be
TCP or UDP.
    targetPort: 80 # Port used by a Service to access the target container.
    This port is closely related to the applications running in a container. In this
    example, the Nginx image uses port 80 by default.
  selector: # Label selector. A Service selects a pod based on the
    label and forwards the requests for accessing the Service to the pod. In this
    example, select the pod with the app:nginx label.
    app: nginx
  type: NodePort # Service type. NodePort indicates that the Service is
    accessed through a node port.
```

**Paso 3** Cree una carga de trabajo.

### kubectl create -f nginx-deployment.yaml

Si se muestra información similar a la siguiente, se ha creado la carga de trabajo.

```
deployment "nginx" created
```

### kubectl get po

Si se muestra la información similar a la siguiente, la carga de trabajo se está ejecutando.

NAME	READY	STATUS	RESTARTS	AGE
nginx-2601814895-qhxqv	1/1	Running	0	9s

## Paso 4 Cree un Service.

### kubectl create -f nginx-nodeport-svc.yaml

Si se muestra información similar a la siguiente, se está creando el Service.

```
service "nginx-nodeport" created
```

### kubectl get svc

Si se muestra la información similar a la siguiente, se ha creado el Service.

```
# kubectl get svc
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.247.0.1	<none>	443/TCP	4d8h
nginx-nodeport	NodePort	10.247.30.40	<none>	8080:30000/TCP	18s

## Paso 5 Acceda al Servicio.

De forma predeterminada, se puede acceder a un Service de NodePort con *Any node IP address:Node port*.

Se puede acceder al Service desde un nodo en otro clúster en la misma VPC o en otro pod del clúster. Si una dirección IP pública está vinculada al nodo, también puede usar la dirección IP pública para acceder al Service. Cree un contenedor en el clúster y acceda al contenedor con *Node IP address:Node port*.

```
# kubectl get node -owide
```

NAME	STATUS	ROLES	AGE	INTERNAL-IP	EXTERNAL-IP	OS-
10.100.0.136	Ready	<none>	152m	10.100.0.136	<none>	CentOS Linux
7 (Core)	3.10.0-1160.25.1.el7.x86_64			docker://18.9.0		
10.100.0.5	Ready	<none>	152m	10.100.0.5	<none>	CentOS Linux
7 (Core)	3.10.0-1160.25.1.el7.x86_64			docker://18.9.0		

```
# kubectl run -i --tty --image nginx:alpine test --rm /bin/sh
If you do not see a command prompt, try pressing Enter.
/ # curl 10.100.0.136:30000
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
```

```
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
/ #
```

----Fin

## 7.3.4 LoadBalancer

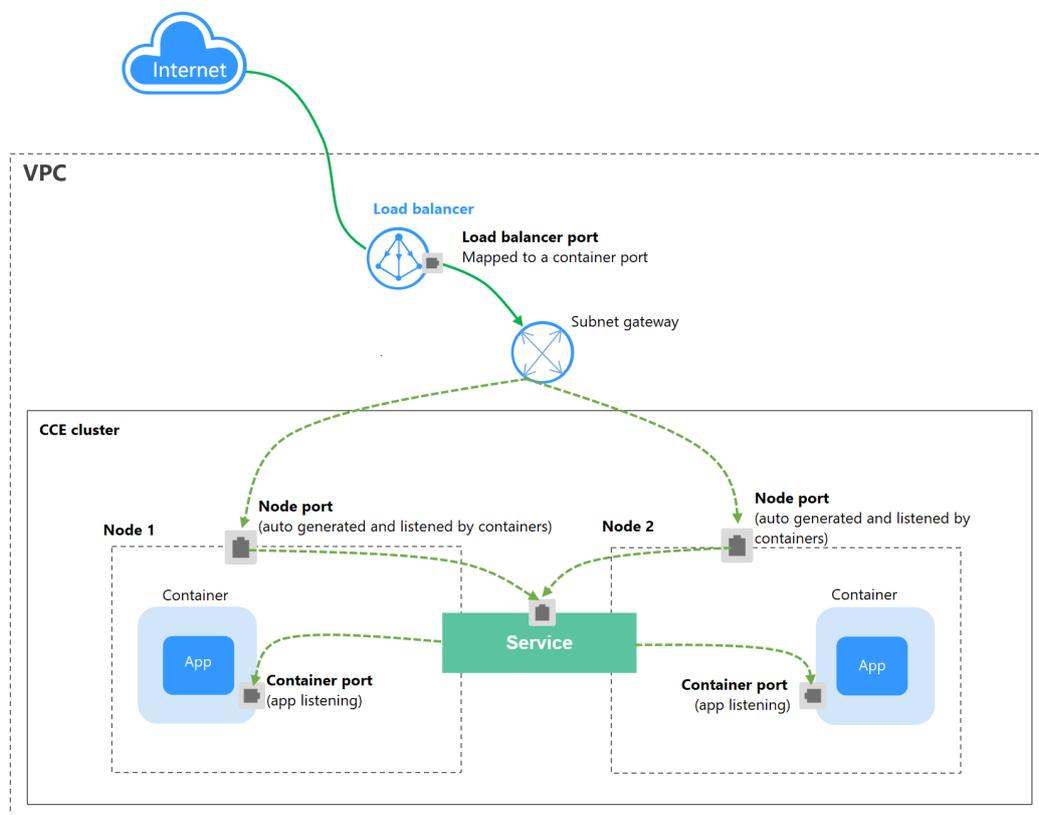
### 7.3.4.1 Creación de un LoadBalancer Service

#### Escenario

Los LoadBalancer Service pueden acceder a cargas de trabajo desde la red pública con ELB, que es más fiable que el acceso basado en EIP. La dirección de acceso de LoadBalancer tiene el formato de *Dirección IP del balanceador de carga de red pública:Puerto de acceso*, por ejemplo, **10.117.117.117:80**.

En este modo de acceso, las solicitudes se transmiten a través de un balanceador de carga de ELB a un nodo y luego se reenvían al pod de destino a través del Servicio.

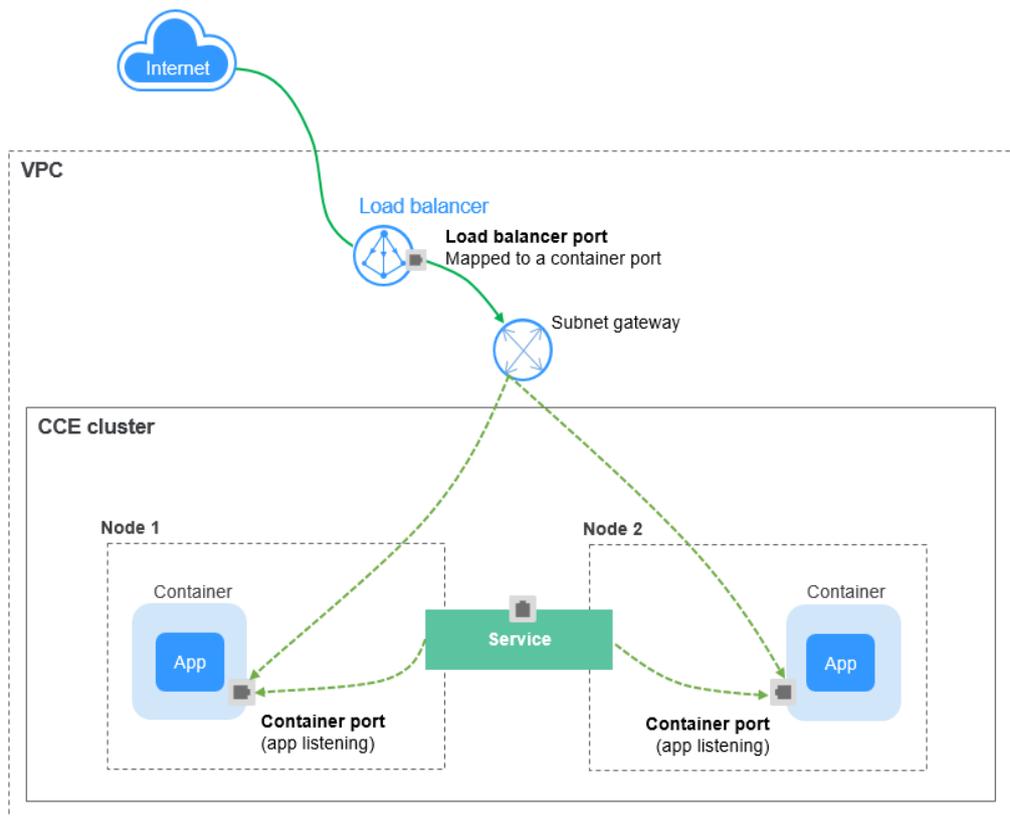
Figura 7-21 LoadBalancer



Cuando se utilizan **clúster de CCE Turbo y balanceador de carga dedicados**, se admite la red de paso a través para reducir la latencia del servicio y garantizar cero pérdidas de rendimiento.

Las solicitudes de acceso externo se reenvían directamente desde un balanceador de carga a los pods. Las solicitudes de acceso interno se pueden reenviar a un pod con un Service.

**Figura 7-22** Redes passthrough



## Restricciones

- LoadBalancer Services permiten acceder a cargas de trabajo desde las redes públicas con ELB. Este modo de acceso tiene las siguientes restricciones:
  - Los balanceadores de carga creados automáticamente no deben ser utilizados por otros recursos. De lo contrario, estos balanceadores de carga no se pueden eliminar por completo.
  - No cambie el nombre de oyente para el balanceador de carga en clústeres de v1.15 y anteriores. De lo contrario, no se puede acceder al balanceador de carga.
- Después de crear un Service, si la configuración de afinidad se cambia del nivel de clúster al nivel de nodo, la tabla de seguimiento de conexiones no se borrará. Se recomienda no modificar la configuración de afinidad del Service después de que se haya creado el Servicio. Si necesita modificarlo, vuelva a crear un Service.
- Si la afinidad de servicio se establece en el nivel de nodo (es decir, se establece **externalTrafficPolicy** en **Local**), el clúster puede no tener acceso al Service mediante la dirección de ELB. Para obtener más información, véase **Por qué un clúster no puede acceder a los servicios mediante el uso de la dirección de ELB**.

- Los clústeres de Turbo de CCE solo admiten la afinidad de servicio a nivel de clúster.
- Los balanceadores de carga de ELB dedicados solo se pueden usar en clústeres de v1.17 y posteriores.
- Los balanceadores de carga dedicados deben ser del tipo de red (TCP/UDP) que admita las redes privadas (con una IP privada). Si Service necesita soportar HTTP, las especificaciones de los balanceadores de carga dedicados deben usar HTTP/HTTPS (balanceo de carga de aplicación) además de TCP/UDP (balanceo de carga de red).
- Si crea un LoadBalancer Service en la consola de CCE, se genera automáticamente un puerto de nodo aleatorio. Si utiliza kubectl para crear un LoadBalancer Service, se genera un puerto de nodo aleatorio a menos que especifique uno.
- En un clúster de CCE, si la afinidad a nivel de clúster está configurada para un LoadBalancer Service, las solicitudes se distribuyen a los puertos de nodo de cada nodo mediante SNAT al ingresar al clúster. El número de puertos de nodo no puede exceder el número de puertos de nodo disponibles en el nodo. Si la afinidad de servicio está en el nivel de nodo (local), no existe tal restricción. En un clúster de CCE Turbo, esta restricción se aplica a los balanceadores de carga ELB compartidos, pero no a los dedicados. Se recomienda utilizar balanceadores de carga ELB dedicados en clústeres de CCE Turbo.
- Cuando el modo de reenvío (proxy) del servicio de clúster es IPVS, la IP del nodo no se puede configurar como la IP externa del Service. De lo contrario, el nodo no está disponible.
- En un clúster que usa el modo proxy IPVS, si el ingreso y el Service usan el mismo balanceador de carga de ELB, no se puede acceder a la entrada desde los nodos y contenedores en el clúster porque kube-proxy monta la dirección de LoadBalancer Service en el puente ipvs-0. Este puente intercepta el tráfico del balanceador de carga conectado a la entrada. Se recomienda utilizar diferentes balanceadores de carga de ELB para la entrada y el Service.

## Creación de un LoadBalancer Service

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** Elija **Networking** en el panel de navegación y haga clic en **Create Service** en la esquina superior derecha.

**Paso 3** Configure los parámetros.

- **Service Name:** Especifique un nombre de Service, que puede ser el mismo que el nombre de la carga de trabajo.
- **Access Type:** Seleccione **LoadBalancer**.
- **Namespace:** Espacio de nombres al que pertenece la carga de trabajo.
- **Service Affinity:** Para más información, véase [externalTrafficPolicy \(afinidad del Service\)](#).
  - **Cluster level:** Las direcciones IP y los puertos de acceso de todos los nodos de un clúster se pueden utilizar para acceder a la carga de trabajo asociada con el Service. El acceso al Service causará una pérdida de rendimiento debido a la redirección de la ruta y no se puede obtener la dirección IP de origen del cliente.
  - **Node level:** Solo la dirección IP y el puerto de acceso del nodo donde se encuentra la carga de trabajo pueden acceder a la carga de trabajo asociada con el Service. El acceso al Service no causará pérdida de rendimiento debido a la redirección de la ruta, y se puede obtener la dirección IP de origen del cliente.

- **Selector:** Agregue una etiqueta y haga clic en **Add**. Un Service selecciona un pod basado en la etiqueta agregada. También puede hacer clic en **Reference Workload Label** para hacer referencia a la etiqueta de una carga de trabajo existente. En el cuadro de diálogo que se muestra, seleccione una carga de trabajo y haga clic en **OK**.
- **IPv6:** Esta función está deshabilitada por defecto. Una vez habilitada esta función, la dirección IP del clúster del Service cambia a una dirección IPv6. Para obtener más información, consulte [Creación de un clúster de doble pila IPv4/IPv6 en CCE](#). **Este parámetro solo está disponible en clústeres de v1.15 o posterior con IPv6 habilitado (establecido durante la creación del clúster).**

- **Load Balancer**

Seleccione el balanceador de carga que desea interconectar. Solo se admiten balanceadores de carga en la misma VPC que el clúster. Si no hay ningún balanceador de carga disponible, haga clic en **Create Load Balancer** para crear uno en la consola de ELB.

La consola de CCE admite la creación automática de balanceadores de carga. Seleccione **Auto create** en el cuadro de lista desplegable y establezca los siguientes parámetros:

- **Instance Name:** Ingrese un nombre de balanceador de carga.
- **Public Access:** Si está habilitado, se creará una EIP con ancho de banda de 5 Mbit/s. De forma predeterminada, se cobra por el tráfico.
- **AZ, Subnet y Specifications** (disponible solo para balanceadores de carga dedicados): Establezca la AZ, la subred y las especificaciones. Actualmente, solo se pueden crear automáticamente los balanceadores de carga dedicados del tipo de red (TCP/UDP).

Puede hacer clic en **Edit** y configurar los parámetros del balanceador de carga en el cuadro de diálogo **Load Balancer**.

- **Distribution Policy:** Hay tres algoritmos disponibles: Round robin ponderado, algoritmo de conexiones mínimas ponderadas o hash de IP de origen.

#### **NOTA**

- **Round robin ponderado:** las solicitudes se reenvían a diferentes servidores en función de sus pesos, lo que indica el rendimiento del procesamiento del servidor. Los servidores backend con mayores pesos reciben proporcionalmente más solicitudes, mientras que los servidores con igual peso reciben el mismo número de solicitudes. Este algoritmo se utiliza a menudo para las conexiones cortas, como los servicios de HTTP.
  - **Conexiones mínimas ponderadas:** Además del peso asignado a cada servidor, también se considera el número de conexiones procesadas por cada servidor backend. Las solicitudes se reenvían al servidor con la relación de conexiones/peso más baja. Basado en las **conexiones mínimas**, el algoritmo **conexiones mínimas ponderadas** asigna un peso a cada servidor basado en su capacidad de procesamiento. Este algoritmo se utiliza a menudo para las conexiones persistentes, tales como las conexiones de base de datos.
  - **Hash de IP de origen:** La dirección IP de origen de cada solicitud se calcula usando el algoritmo hash para obtener una clave hash única, y todos los servidores backend están numerados. La clave generada asigna el cliente a un servidor determinado. Esto permite que las solicitudes de diferentes clientes se distribuyan en modo de equilibrio de carga y garantiza que las solicitudes del mismo cliente se reenvíen al mismo servidor. Este algoritmo se aplica a las conexiones de TCP sin cookies.
- **Type:** Esta función está deshabilitada por defecto. Puede seleccionar **Source IP address**. La sesión adhesiva basada en la dirección IP de origen significa que las solicitudes de acceso desde la misma dirección IP se reenvían al mismo servidor backend.

 **NOTA**

Cuando la **política de distribución** utiliza el algoritmo de dirección IP de origen, no se puede establecer la sesión adhesiva.

- **Health Check:** Establezca la configuración de comprobación de estado del balanceador de carga. De forma predeterminada, la comprobación de estado se realiza globalmente.

Parámetro	Descripción
Protocol	Cuando el protocolo de <b>Puerto</b> se establece en TCP, se admiten TCP y HTTP. Cuando el protocolo del <b>Puerto</b> se establece en UDP, se admite el UDP. <ul style="list-style-type: none"> <li>– <b>Check Path</b> (soportado solo por HTTP): especifica el URL de comprobación de estado. La ruta de comprobación debe comenzar con un (/) de barra diagonal y contener entre 1 y 80 caracteres.</li> </ul>
Port	De forma predeterminada, el puerto de Service (Puerto de nodo y puerto contenedor del Service) se utiliza para la comprobación de estado. También puede especificar otro puerto para la comprobación de estado. Después de especificar el puerto, se agregará un puerto de servicio llamado <b>cce-healthz</b> para el Service. <ul style="list-style-type: none"> <li>– <b>Node Port:</b> si se utiliza un balanceador de carga compartido o no se asocia ninguna instancia ENI, el puerto de nodo se utiliza como puerto de comprobación de estado. Si no se especifica este parámetro, se utiliza un puerto aleatorio. El valor oscila entre 30000 y 32767.</li> <li>– <b>Container Port:</b> Cuando un balanceador de carga dedicado está asociado a una instancia ENI, el puerto contenedor se utiliza para la comprobación de estado. El valor varía de 1 a 65535.</li> </ul>
Check Period (s)	Especifica el intervalo máximo entre las comprobaciones de estado. El valor varía de 1 a 50.
Timeout (s)	Especifica la duración máxima del tiempo de espera para cada comprobación de estado. El valor varía de 1 a 50.
Max. Retries	Especifica el número máximo de reintentos de comprobación de estado. El valor varía de 1 a 10.

- **Port**
  - **Protocol:** protocolo utilizado por el Service.
  - **Service Port:** puerto utilizado por el Service. El número de puerto se encuentra dentro del rango de 1 a 65535.
  - **Container Port:** puerto en el que escucha la carga de trabajo. Por ejemplo, Nginx utiliza el puerto 80 de forma predeterminada.
- **Annotation:** El LoadBalancer Service tiene algunas funciones avanzadas de CCE, que se implementan mediante anotaciones. Para obtener más información, véase **Uso de anotaciones para configurar el balanceo de carga**.

**Paso 4** Haga clic en **OK**.

----Fin

## Uso de kubectl para crear un Service (Uso de un balanceador de carga existente)

Puede establecer el Service al crear una carga de trabajo con kubectl. En esta sección se utiliza una carga de trabajo Nginx como ejemplo para describir cómo agregar un LoadBalancer Service usando kubectl.

**Paso 1** Utilice kubectl para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).

**Paso 2** Cree los archivos llamados **nginx-deployment.yaml** y **nginx-elb-svc.yaml** y edítelos.

Los nombres de archivo están definidos por el usuario. **nginx-despliegue.yaml** y **nginx-elb-svc.yaml** son simplemente los nombres de archivo de ejemplo.

### vi nginx-deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx
        name: nginx
      imagePullSecrets:
      - name: default-secret
```

### vi nginx-elb-svc.yaml

#### NOTA

Antes de habilitar la sesión adhesiva, asegúrese de que se cumplen las siguientes condiciones:

- El protocolo de carga de trabajo es TCP.
- La antiafinidad se ha configurado entre los pods de la carga de trabajo. Es decir, todos los pods de la carga de trabajo se despliegan en los nodos diferentes. Para obtener más información, véase [Política de programación \(afinidad/antiafinidad\)](#).

```
apiVersion: v1
kind: Service
metadata:
  name: nginx
  annotations:
    kubernetes.io/elb.id: <your_elb_id> # ELB ID. Replace
it with the actual value.
    kubernetes.io/elb.class: performance # Load balancer type
    kubernetes.io/elb.lb-algorithm: ROUND_ROBIN # Load balancer
algorithm
    kubernetes.io/elb.session-affinity-mode: SOURCE_IP # The sticky
session type is source IP address.
    kubernetes.io/elb.session-affinity-option: '{"persistence_timeout":
"30"}' # Stickiness duration (min)
    kubernetes.io/elb.health-check-flag: 'on' # Enable the ELB
health check function.
    kubernetes.io/elb.health-check-option: '{
  "protocol": "TCP",
  "delay": "5",
```

```

        "timeout": "10",
        "max_retries": "3"
    }'
spec:
  selector:
    app: nginx
  ports:
  - name: service0
    port: 80 # Port for accessing the Service, which is also the listener
    port on the load balancer.
    protocol: TCP
    targetPort: 80 # Port used by a Service to access the target container. This
    port is closely related to the applications running in a container.
    type: LoadBalancer
    
```

El ejemplo anterior utiliza anotaciones para implementar algunas funciones avanzadas de balanceo de carga, como la sesión adhesiva y la comprobación de estado. Para obtener más información, véase [Tabla 7-4](#).

Además de las funciones de este ejemplo, para más anotaciones y ejemplos relacionados con funciones avanzadas, consulte [Uso de anotaciones para configurar el balanceo de carga](#).

**Tabla 7-4** parámetros de anotaciones

Parámetro	Obligatorio	Tipo	Descripción
kubernetes.io/elb.id	Sí	String	<p>ID de un balanceador de carga mejorado. Obligatorio cuando se va a asociar un balanceador de carga existente.</p> <p><b>Cómo obtenerlo:</b></p> <p>En la consola de gestión, haga clic en <b>Service List</b> y elija <b>Networking &gt; Elastic Load Balance</b>. Haga clic en el nombre del balanceador de carga de destino. En la página de ficha <b>Summary</b>, encuentre y copie el ID.</p> <p><b>NOTA</b></p> <p>El sistema se conecta preferentemente al balanceador de carga basándose en el campo <b>kubernetes.io/elb.id</b>. Si no se especifica este campo, se utiliza el campo <b>spec.loadBalancerIP</b> (opcional y disponible solo en 1.23 y versiones anteriores).</p> <p>No utilice el campo <b>spec.loadBalancerIP</b> para conectarse al balanceador de carga. Este campo será descartado por Kubernetes. Para obtener más información, consulte <a href="#">Deprecation</a>.</p>

Parámetro	Obligatorio	Tipo	Descripción
kubernetes.io/elb.class	Sí	String	<p>Seleccione un tipo de balanceador de carga adecuado.</p> <p>El valor puede ser:</p> <ul style="list-style-type: none"> <li>● <b>union</b>: balanceador de carga compartido</li> <li>● <b>performance</b>: balanceador de carga dedicado, que solo se puede utilizar en clústeres de v1.17 y posteriores. Para obtener más información, consulte <a href="#">Diferencias entre los balanceadores de carga compartidos y los dedicados</a>.</li> </ul>
kubernetes.io/elb.lb-algorithm	No	String	<p>Especifica el algoritmo de equilibrio de carga del grupo de servidores backend. El valor predeterminado es <b>ROUND_ROBIN</b>.</p> <p>Opciones:</p> <ul style="list-style-type: none"> <li>● <b>ROUND_ROBIN</b>: algoritmo de round robin ponderado</li> <li>● <b>LEAST_CONNECTIONS</b>: algoritmo de conexiones mínimas ponderadas</li> <li>● <b>SOURCE_IP</b>: algoritmo de hash IP de origen</li> </ul> <p><b>NOTA</b>                      Si este parámetro se establece en <b>SOURCE_IP</b>, la configuración de ponderación (campo <b>weight</b>) de los servidores backend vinculados al grupo de servidores backend no es válida y no se puede habilitar la sesión adhesiva.</p>
kubernetes.io/elb.session-affinity-mode	No	String	<p>Se admite la sesión adhesiva basada en la dirección IP de origen. Es decir, las solicitudes de acceso desde la misma dirección IP se reenvían al mismo servidor backend.</p> <ul style="list-style-type: none"> <li>● Deshabilitar sesión adhesiva: No configure este parámetro.</li> <li>● Activar la sesión adhesiva: Establezca este parámetro en <b>SOURCE_IP</b> para indicar que la sesión adhesiva se basa en la dirección IP de origen.</li> </ul> <p><b>NOTA</b>                      Cuando <b>kubernetes.io/elb.lb-algorithm</b> se establece en <b>SOURCE_IP</b> (algoritmo de dirección IP de origen), no se puede activar la sesión adhesiva.</p>

Parámetro	Obligatorio	Tipo	Descripción
kubernetes.io/elb.session-affinity-option	No	<a href="#">Tabla 7-5</a> object	Tiempo de espera de sesión adhesiva.
kubernetes.io/elb.health-check-flag	No	String	<p>Si se debe habilitar la comprobación de estado del ELB.</p> <ul style="list-style-type: none"> <li>● Habilitar la comprobación de estado: Deje en blanco este parámetro o configúrelo en <b>on</b>.</li> <li>● Deshabilitar la comprobación de estado: Establezca este parámetro en <b>off</b>.</li> </ul> <p>Si este parámetro está habilitado, el campo <a href="#">kubernetes.io/elb.health-check-option</a> también debe especificarse al mismo tiempo.</p>
kubernetes.io/elb.health-check-option	No	<a href="#">Tabla 7-6</a> object	Elementos de configuración de comprobación de estado de ELB.

**Tabla 7-5** Estructura de datos del campo **elb.session-affinity-option**

Parámetro	Obligatorio	Tipo	Descripción
persistence_timeout	Sí	String	<p>Tiempo de espera de sesión adhesiva, en minutos. Este parámetro solo es válido cuando <b>elb.session-affinity-mode</b> está establecido en <b>SOURCE_IP</b>.</p> <p>Rango de valores: 1 a 60. Valor predeterminado: <b>60</b></p>

**Tabla 7-6** Descripción de la estructura de datos del campo **elb.health-check-option**

Parámetro	Obligatorio	Tipo	Descripción
delay	No	String	<p>Tiempo de espera inicial (en segundos) para iniciar la comprobación de estado.</p> <p>Rango de valores: 1 a 50. Valor predeterminado: <b>5</b></p>

Parámetro	Obligatorio	Tipo	Descripción
timeout	No	String	Tiempo de espera de la comprobación de estado, en segundos. Rango de valores: 1 a 50. Valor predeterminado: <b>10</b>
max_retries	No	String	Número máximo de reintentos de comprobación de estado. Rango de valores: 1 a 10. Valor predeterminado: <b>3</b>
protocol	No	String	Protocolo de comprobación de estado. Opciones de valor: TCP o HTTP
path	No	String	URL de comprobación de estado. Este parámetro debe configurarse cuando el protocolo es HTTP. Valor predeterminado: / El valor puede contener de 1 a 10,000 caracteres.

**Paso 3** Cree una carga de trabajo.

**kubectl create -f nginx-deployment.yaml**

Si se muestra información similar a la siguiente, se ha creado la carga de trabajo.

```
deployment/nginx created
```

**kubectl get pod**

Si se muestra la información similar a la siguiente, la carga de trabajo se está ejecutando.

NAME	READY	STATUS	RESTARTS	AGE
<b>nginx-2601814895-c1xhw</b>	<b>1/1</b>	<b>Running</b>	<b>0</b>	<b>6s</b>

**Paso 4** Crear un Service.

**kubectl create -f nginx-elb-svc.yaml**

Si se muestra la información similar a la siguiente, se ha creado el Service.

```
service/nginx created
```

**kubectl get svc**

Si se muestra la información similar a la siguiente, se ha definido el tipo de acceso y se puede acceder a la carga de trabajo.

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.247.0.1	<none>	443/TCP	3d
<b>nginx</b>	<b>LoadBalancer</b>	<b>10.247.130.196</b>	<b>10.78.42.242</b>	<b>80:31540/TCP</b>	<b>51s</b>

**Paso 5** Ingrese el URL en el cuadro de dirección del navegador, por ejemplo, **10.78.42.242:80**.

**10.78.42.242** indica la dirección IP del balanceador de carga y **80** indica el puerto de acceso que se muestra en la consola de CCE.

El Nginx es accesible.

**Figura 7-23** Acceso a Nginx a través del LoadBalancer Service

## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](https://nginx.org).  
Commercial support is available at [nginx.com](https://nginx.com).

*Thank you for using nginx.*

----Fin

## Uso de kubectl para crear un Service (creación automática de un balanceador de carga)

Puede establecer el Service al crear una carga de trabajo con kubectl. En esta sección se utiliza una carga de trabajo Nginx como ejemplo para describir cómo agregar un LoadBalancer Service usando kubectl.

- Paso 1** Utilice kubectl para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).
- Paso 2** Cree los archivos llamados **nginx-deployment.yaml** y **nginx-elb-svc.yaml** y edítelos.

Los nombres de archivo están definidos por el usuario. **nginx-despliegue.yaml** y **nginx-elb-svc.yaml** son simplemente los nombres de archivo de ejemplo.

### vi nginx-deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx
        name: nginx
        imagePullSecrets:
        - name: default-secret
```

### vi nginx-elb-svc.yaml

 **NOTA**

Antes de habilitar la sesión adhesiva, asegúrese de que se cumplen las siguientes condiciones:

- El protocolo de carga de trabajo es TCP.
- La antiafinidad se ha configurado entre los pods de la carga de trabajo. Es decir, todos los pods de la carga de trabajo se despliegan en diferentes nodos. Para obtener más información, véase [Política de programación \(afinidad/antiafinidad\)](#).

Ejemplo de un Service que utiliza un balanceador de carga compartido de red pública:

```

apiVersion: v1
kind: Service
metadata:
  annotations:
    kubernetes.io/elb.class: union
    kubernetes.io/elb.autocreate: '{
      "type": "public",
      "bandwidth_name": "cce-bandwidth-1551163379627",
      "bandwidth_chargemode": "bandwidth",
      "bandwidth_size": 5,
      "bandwidth_sharetype": "PER",
      "eip_type": "5_bgp"
    }'
    kubernetes.io/elb.enterpriseID: 0 # ID of the enterprise project to
which the load balancer belongs
    kubernetes.io/elb.lb-algorithm: ROUND_ROBIN # Load balancer
algorithm
    kubernetes.io/elb.session-affinity-mode: SOURCE_IP # The sticky
session type is source IP address.
    kubernetes.io/elb.session-affinity-option: '{"persistence_timeout":
"30"}' # Stickiness duration (min)
    kubernetes.io/elb.health-check-flag: 'on' # Enable the ELB
health check function.
    kubernetes.io/elb.health-check-option: '{
      "protocol": "TCP",
      "delay": "5",
      "timeout": "10",
      "max_retries": "3"
    }'
  labels:
    app: nginx
    name: nginx
spec:
  ports:
    - name: service0
      port: 80
      protocol: TCP
      targetPort: 80
  selector:
    app: nginx
  type: LoadBalancer
    
```

Ejemplo de Service que utiliza un balanceador de carga dedicado de red pública (solo para clústeres de v1.17 y versiones posteriores):

```

apiVersion: v1
kind: Service
metadata:
  name: nginx
  labels:
    app: nginx
  namespace: default
  annotations:
    kubernetes.io/elb.class: performance
    kubernetes.io/elb.autocreate: '{
      "type": "public",
      "bandwidth_name": "cce-bandwidth-1626694478577",
      "bandwidth_chargemode": "bandwidth",
      "bandwidth_size": 5,
    }'
  
```

```

        "bandwidth_sharetype": "PER",
        "eip_type": "5_bgp",
        "available_zone": [
            ""
        ],
        "l4_flavor_name": "L4_flavor.elb.s1.small"
    }'
    kubernetes.io/elb.enterpriseID: 0          # ID of the enterprise project to
which the load balancer belongs
    kubernetes.io/elb.lb-algorithm: ROUND_ROBIN          # Load balancer
algorithm
    kubernetes.io/elb.session-affinity-mode: SOURCE_IP          # The sticky
session type is source IP address.
    kubernetes.io/elb.session-affinity-option: '{"persistence_timeout":
"30"}'          # Stickiness duration (min)
    kubernetes.io/elb.health-check-flag: 'on'          # Enable the ELB
health check function.
    kubernetes.io/elb.health-check-option: '{
        "protocol": "TCP",
        "delay": "5",
        "timeout": "10",
        "max_retries": "3"
    }'
spec:
  selector:
    app: nginx
  ports:
  - name: cce-service-0
    targetPort: 80
    nodePort: 0
    port: 80
    protocol: TCP
  type: LoadBalancer
    
```

El ejemplo anterior utiliza anotaciones para implementar algunas funciones avanzadas de balanceo de carga, como la sesión adhesiva y la comprobación de estado. Para obtener más información, véase [Tabla 7-7](#).

Además de las funciones de este ejemplo, para más anotaciones y ejemplos relacionados con funciones avanzadas, consulte [Uso de anotaciones para configurar el balanceo de carga](#).

**Tabla 7-7** Parámetros de anotaciones

Parámetro	Obligatorio	Tipo	Descripción
kubernetes.io/elb.class	Sí	String	Seleccione un tipo de balanceador de carga adecuado. El valor puede ser: <ul style="list-style-type: none"> <li>● <b>union</b>: balanceador de carga compartido</li> <li>● <b>performance</b>: balanceador de carga dedicado, que solo se puede utilizar en clústeres de v1.17 y posteriores. Para obtener más información, consulte <a href="#">Diferencias entre los balanceadores de carga compartidos y los dedicados</a>.</li> </ul>

Parámetro	Obligatorio	Tipo	Descripción
kubernetes.io/ elb.autocreate	Sí	<b>elb.autocreate</b> object	<p>Si se debe crear automáticamente un balanceador de carga asociado con el Service.</p> <p><b>Ejemplo</b></p> <ul style="list-style-type: none"> <li>● Si se creará automáticamente un balanceador de carga de red pública, establezca este parámetro en el siguiente valor:  <pre>{"type":"public","bandwidth_name":"cce-bandwidth-1551163379627","bandwidth_chargemode":"bandwidth","bandwidth_size":5,"bandwidth_sharetype":"PER","eip_type":"5_bgp","name":"james"}</pre> </li> <li>● Si se creará automáticamente un balanceador de carga de red privada, establezca este parámetro en el siguiente valor:  <pre>{"type":"inner","name":"A-location-d-test"}</pre> </li> </ul>
kubernetes.io/ elb.subnet-id	-	String	<p>ID de la subred donde se encuentra el clúster. El valor puede contener de 1 a 100 caracteres.</p> <ul style="list-style-type: none"> <li>● Obligatorio cuando se va a crear automáticamente un clúster de v1.11.7-r0 o anterior.</li> <li>● Opcional para los clústeres posteriores a v1.11.7-r0.</li> </ul> <p>Para obtener más detalles sobre cómo obtener el valor, consulte <a href="#">¿Cuál es la diferencia entre la API de subred de VPC y la API de subred de OpenStack Neutron?</a></p>

Parámetro	Obligatorio	Tipo	Descripción
kubernetes.io/ elb.enterpriseID	No	String	<p><b>Los clústeres de v1.15 y versiones posteriores admiten este campo. En clústeres anteriores a v1.15, los balanceadores de carga se crean en el proyecto predeterminado de forma predeterminada.</b></p> <p>Este parámetro indica el ID del proyecto de empresa en el que se creará el balanceador de carga de ELB.</p> <p>Si este parámetro no se especifica o se establece en <b>0</b>, los recursos estarán enlazados al proyecto de empresa predeterminado.</p> <p><b>Cómo obtenerlo:</b></p> <p>Inicie sesión en la consola de gestión y seleccione <b>Enterprise &gt; Project Management</b> en la barra de menú superior. En la lista que se muestra, haga clic en el nombre del proyecto de empresa de destino y copie el ID en la página de detalles del proyecto de empresa.</p>
kubernetes.io/elb.lb- algorithm	No	String	<p>Especifica el algoritmo de equilibrio de carga del grupo de servidores backend. El valor predeterminado es <b>ROUND_ROBIN</b>.</p> <p>Opciones:</p> <ul style="list-style-type: none"> <li>● <b>ROUND_ROBIN</b>: algoritmo de round robin ponderado</li> <li>● <b>LEAST_CONNECTIONS</b>: algoritmo de conexiones mínimas ponderadas</li> <li>● <b>SOURCE_IP</b>: algoritmo de hash IP de origen</li> </ul> <p><b>NOTA</b></p> <p>Si este parámetro se establece en <b>SOURCE_IP</b>, la configuración de ponderación (campo <b>weight</b>) de los servidores backend vinculados al grupo de servidores backend no es válida y no se puede habilitar la sesión adhesiva.</p>

Parámetro	Obligatorio	Tipo	Descripción
kubernetes.io/ elb.session-affinity- mode	No	String	<p>Se admite la sesión adhesiva basada en la dirección IP de origen. Es decir, las solicitudes de acceso desde la misma dirección IP se reenvían al mismo servidor backend.</p> <ul style="list-style-type: none"> <li>● Deshabilitar sesión adhesiva: No configure este parámetro.</li> <li>● Activar la sesión adhesiva: Establezca este parámetro en <b>SOURCE_IP</b> para indicar que la sesión adhesiva se basa en la dirección IP de origen.</li> </ul> <p><b>NOTA</b>                      Cuando <b>kubernetes.io/elb.lb-algorithm</b> se establece en <b>SOURCE_IP</b> (algoritmo de dirección IP de origen), no se puede activar la sesión adhesiva.</p>
kubernetes.io/ elb.session-affinity- option	No	<b>Tabla 7-5</b> object	Tiempo de espera de sesión adhesiva.
kubernetes.io/ elb.health-check- flag	No	String	<p>Si se debe habilitar la comprobación de estado del ELB.</p> <ul style="list-style-type: none"> <li>● Habilitar la comprobación de estado: Deje en blanco este parámetro o configúrelo en <b>on</b>.</li> <li>● Deshabilitar la comprobación de estado: Establezca este parámetro en <b>off</b>.</li> </ul> <p>Si este parámetro está habilitado, el campo <b>kubernetes.io/elb.health-check-option</b> también debe especificarse al mismo tiempo.</p>
kubernetes.io/ elb.health-check- option	No	<b>Tabla 7-6</b> object	Elementos de configuración de comprobación de estado de ELB.

**Tabla 7-8** Estructura de datos del campo **elb.autocreate**

Parámetro	Obligatorio	Tipo	Descripción
name	No	String	Nombre del balanceador de carga creado automáticamente. Rango de valores: de 1 a 64 caracteres, incluidas las letras minúsculas, los dígitos y los guiones bajos (_). El valor debe comenzar con una letra minúscula y terminar con una letra minúscula o un dígito. Predeterminado: <b>cce-lb+service.UID</b>
type	No	String	Tipo de red del balanceador de carga. <ul style="list-style-type: none"> <li>● <b>public</b>: balanceador de carga de red pública</li> <li>● <b>inner</b>: balanceador de carga de red privada</li> </ul> Predeterminado: <b>inner</b>
bandwidth_name	Sí para los balanceadores de carga de red pública	String	Nombre del ancho de banda. El valor predeterminado es <b>cce-bandwidth-*****</b> . Rango de valores: de 1 a 64 caracteres, incluidas las letras minúsculas, los dígitos y los guiones bajos (_). El valor debe comenzar con una letra minúscula y terminar con una letra minúscula o un dígito.
bandwidth_chargemode	No	String	Modo de facturación de ancho de banda. <ul style="list-style-type: none"> <li>● <b>bandwidth</b>: facturado por ancho de banda</li> <li>● <b>traffic</b>: facturado por tráfico</li> </ul> Predeterminado: <b>bandwidth</b>
bandwidth_size	Sí para los balanceadores de carga de red pública	Integer	Tamaño del ancho de banda. El valor predeterminado es de 1 a 2000 Mbit/s. Configure este parámetro en función del rango de ancho de banda permitido en su región.
bandwidth_sharetype	Sí para los balanceadores de carga de red pública	String	Modo de uso compartido de ancho de banda. <ul style="list-style-type: none"> <li>● <b>PER</b>: Ancho de banda dedicado</li> </ul>

Parámetro	Obligatorio	Tipo	Descripción
eip_type	Sí para los balanceadores de carga de red pública	String	Tipo de la EIP. <ul style="list-style-type: none"> <li>● <b>5_telcom</b>: China Telecom</li> <li>● <b>5_union</b>: China Unicom</li> <li>● <b>5_bgp</b>: BGP dinámico</li> <li>● <b>5_sbgp</b>: BGP estático</li> </ul>
vip_subnet_cidr_id	No	String	Subred donde se encuentra el balanceador de carga. Este campo es compatible con clústeres de v1.21 o posterior. Si no se especifica este parámetro, el balanceador de carga de ELB y el clúster están en la misma subred.
available_zone	Sí	Array of strings	AZ donde se encuentra el balanceador de carga. Puede obtener todas las AZ soportadas por <a href="#">consultar la lista de AZ</a> . Este parámetro solo está disponible para los balanceadores de carga dedicados.
l4_flavor_name	Sí	String	Nombre de la variante del balanceador de carga de capa 4. Puede obtener todos los tipos admitidos <a href="#">consultando la lista de variantes</a> . Este parámetro solo está disponible para los balanceadores de carga dedicados.
l7_flavor_name	No	String	Nombre de la variante del balanceador de carga de capa 7. Puede obtener todos los tipos admitidos <a href="#">consultando la lista de variantes</a> . Este parámetro solo está disponible para los balanceadores de carga dedicados. El valor de este parámetro debe ser el mismo que el de <b>l4_flavor_name</b> , es decir, ambas son especificaciones elásticas o especificaciones fijas.

Parámetro	Obligatorio	Tipo	Descripción
elb_virsubnet_ids	No	Array of strings	<p>Subred donde se encuentra el servidor de backend del balanceador de carga. Si este parámetro se deja en blanco, se utiliza la subred de clúster predeterminada. Los balanceadores de carga ocupan un número diferente de direcciones IP de subred según sus especificaciones. Por lo tanto, no se recomienda utilizar los bloques CIDR de subred de otros recursos (como clústeres y nodos) como el bloque CIDR del balanceador de carga.</p> <p>Este parámetro solo está disponible para los balanceadores de carga dedicados.</p> <p>Por ejemplo:</p> <pre>"elb_virsubnet_ids": [   "14567f27-8ae4-42b8-ae47-9f847a4690dd" ]</pre>

**Paso 3** Cree una carga de trabajo.

**kubectl create -f nginx-deployment.yaml**

Si se muestra la información similar a la siguiente, se está creando la carga de trabajo.

```
deployment/nginx created
```

**kubectl get pod**

Si se muestra la información similar a la siguiente, la carga de trabajo se está ejecutando.

NAME	READY	STATUS	RESTARTS	AGE
nginx-2601814895-c1xhw	1/1	Running	0	6s

**Paso 4** Crear un Service.

**kubectl create -f nginx-elb-svc.yaml**

Si se muestra la información similar a la siguiente, se ha creado el Service.

```
service/nginx created
```

**kubectl get svc**

Si se muestra la información similar a la siguiente, se ha definido el tipo de acceso y se puede acceder a la carga de trabajo.

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.247.0.1	<none>	443/TCP	3d
nginx	LoadBalancer	10.247.130.196	10.78.42.242	80:31540/TCP	51s

**Paso 5** Ingrese el URL en el cuadro de dirección del navegador, por ejemplo, **10.78.42.242:80**.

**10.78.42.242** indica la dirección IP del balanceador de carga y **80** indica el puerto de acceso que se muestra en la consola de CCE.

El Nginx es accesible.

**Figura 7-24** Acceso a Nginx a través del LoadBalancer Service

# Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
 Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

---Fin

## Por qué un clúster no puede acceder a los servicios mediante el uso de la dirección de ELB

Si la afinidad de servicio de un Servicio de LoadBalancer se establece en el nivel de nodo, es decir, el valor de **externalTrafficPolicy** es **Local**, es posible que no se pueda acceder a la dirección de ELB desde el clúster (específicamente, los nodos o contenedores). La información que aparecerá en pantalla será similar a la información siguiente:

```
upstream connect error or disconnect/reset before headers. reset reason:
connection failure
```

Esto se debe a que cuando se crea el LoadBalancer Service, kube-proxy agrega la dirección de acceso de ELB como IP externa a iptables o IPVS. Si un cliente inicia una solicitud para acceder a la dirección de ELB desde dentro del clúster, la dirección se considera la dirección IP externa del Service y se reenvía directamente por kube-proxy sin pasar a través del ELB fuera del clúster.

Cuando el valor de **externalTrafficPolicy** es **Local**, la situación varía según el modelo de red contenedor y el modo de reenvío de Service. Los detalles son los siguientes:

Servidor	Cliente	Clúster de red de túneles (IPVS)	Clúster de red de VPC (IPVS)	Clúster de red de túneles (iptables)	Clúster de red de VPC (iptables)
NodePort Service	El mismo nodo	OK. El nodo donde se ejecuta el pod es accesible, no cualquier otro nodo.	OK. El nodo donde se ejecuta el pod es accesible.	OK. El nodo donde se ejecuta el pod es accesible.	OK. El nodo donde se ejecuta el pod es accesible.

	Entre los nodos	OK. El nodo donde se ejecuta el pod es accesible, no cualquier otro nodo.	OK. El nodo donde se ejecuta el pod es accesible.	OK. El nodo donde se ejecuta el pod es accesible visitando la IP + el puerto del nodo, no de ninguna otra manera.	OK. El nodo donde se ejecuta el pod es accesible visitando la IP + el puerto del nodo, no de ninguna otra manera.
	Contenedores en el mismo nodo	OK. El nodo donde se ejecuta el pod es accesible, no cualquier otro nodo.	OK. El nodo donde se ejecuta el pod no es accesible.	OK. El nodo donde se ejecuta el pod es accesible.	OK. El nodo donde se ejecuta el pod no es accesible.
	Contenedores entre los nodos	OK. El nodo donde se ejecuta el pod es accesible, no cualquier otro nodo.	OK. El nodo donde se ejecuta el pod es accesible.	OK. El nodo donde se ejecuta el pod es accesible.	OK. El nodo donde se ejecuta el pod es accesible.
LoadBalancer Service que utiliza un balanceador de carga dedicado	El mismo nodo	Accesible para las redes públicas, no para las redes privadas.	Accesible para las redes públicas, no para las redes privadas.	Accesible para las redes públicas, no para las redes privadas.	Accesible para las redes públicas, no para las redes privadas.
	Contenedores en el mismo nodo	Accesible para las redes públicas, no para las redes privadas.	Accesible para las redes públicas, no para las redes privadas.	Accesible para las redes públicas, no para las redes privadas.	Accesible para las redes públicas, no para las redes privadas.
Service local del complemento nginx-ingress mediante un balanceador de carga dedicado	El mismo nodo	Accesible para las redes públicas, no para las redes privadas.	Accesible para las redes públicas, no para las redes privadas.	Accesible para las redes públicas, no para las redes privadas.	Accesible para las redes públicas, no para las redes privadas.

	Contenedores en el mismo nodo	Accesible para las redes públicas, no para las redes privadas.	Accesible para las redes públicas, no para las redes privadas.	Accesible para las redes públicas, no para las redes privadas.	Accesible para las redes públicas, no para las redes privadas.
--	-------------------------------	--	--	--	--

Se pueden utilizar los siguientes métodos para resolver este problema:

- **(Recomendado)** En el clúster, utilice el nombre de dominio de servicio o servicio de ClusterIP para el acceso.
- Establezca **externalTrafficPolicy** del servicio en **Cluster** es decir, la afinidad de servicio a nivel de clúster. Tenga en cuenta que esto afecta a la persistencia de la dirección de origen.

```
apiVersion: v1
kind: Service
metadata:
  annotations:
    kubernetes.io/elb.class: union
    kubernetes.io/elb.autocreate: '{"type":"public","bandwidth_name":"cce-bandwidth","bandwidth_chargemode":"bandwidth","bandwidth_size":5,"bandwidth_sharetype":"PER","eip_type":"5_bgp","name":"james"}'
  labels:
    app: nginx
    name: nginx
spec:
  externalTrafficPolicy: Cluster
  ports:
  - name: service0
    port: 80
    protocol: TCP
    targetPort: 80
  selector:
    app: nginx
  type: LoadBalancer
```

- Aprovechando la función de pass-through del Service, kube-proxy se omite cuando se utiliza la dirección de ELB para acceder. Primero se accede al balanceador de carga de ELB y, a continuación, a la carga de trabajo. Para obtener más información, véase [Habilitación de redes de paso a través para los servicios de LoadBalancer](#).

**NOTA**

- Después de configurar las redes de paso a través para un balanceador de carga dedicado, no se puede acceder a contenedores en el nodo donde se ejecuta la carga de trabajo a través del Service.
- Las redes de paso a través no son compatibles con clústeres de v1.15 o anteriores.
- En el modo de red IPVS, la configuración de paso a través del Service conectado al mismo ELB debe ser la misma.

```
apiVersion: v1
kind: Service
metadata:
  annotations:
    kubernetes.io/elb.pass-through: "true"
    kubernetes.io/elb.class: union
    kubernetes.io/elb.autocreate: '{"type":"public","bandwidth_name":"cce-bandwidth","bandwidth_chargemode":"bandwidth","bandwidth_size":5,"bandwidth_sharetype":"PER","eip_type":"5_bgp","name":"james"}'
```

```

labels:
  app: nginx
  name: nginx
spec:
  externalTrafficPolicy: Local
  ports:
  - name: service0
    port: 80
    protocol: TCP
    targetPort: 80
  selector:
    app: nginx
  type: LoadBalancer
    
```

### 7.3.4.2 Uso de anotaciones para configurar el balanceo de carga

Puede agregar anotaciones a un archivo YAML para usar algunas funciones avanzadas de CCE. En esta sección se describen las anotaciones disponibles cuando se crea un servicio de LoadBalancer.

- [Interconexión con ELB](#)
- [Sesión adhesiva](#)
- [Comprobación de estado](#)
- [Protocolo HTTP](#)
- [Ajuste dinámico de ponderación del backend ECS](#)
- [Capacidad de pass-through](#)
- [Lista blanca](#)
- [Red de host](#)

## Interconexión con ELB

Tabla 7-9 Anotaciones para interconectar con ELB

Parámetro	Tipo	Descripción	Versión de clúster admitida
kubernetes.io/elb.class	String	Seleccione un tipo de balanceador de carga adecuado. El valor puede ser: <ul style="list-style-type: none"> <li>● <b>union</b>: balanceador de carga compartido</li> <li>● <b>performance</b>: balanceador de carga dedicado, que solo se puede utilizar en clústeres de v1.17 y posteriores. Para obtener más información, consulte <a href="#">Diferencias entre los balanceadores de carga compartidos y los dedicados</a>.</li> </ul>	v1.9 o posterior

Parámetro	Tipo	Descripción	Versión de clúster admitida
kubernetes.io/elb.id	String	<p>Obligatorio <b>cuando se va a asociar un balanceador de carga existente</b>.</p> <p>ID de un balanceador de carga.</p> <p><b>Cómo obtenerlo:</b></p> <p>En la consola de gestión, haga clic en <b>Service List</b> y elija <b>Networking &gt; Elastic Load Balance</b>. Haga clic en el nombre del balanceador de carga de destino. En la página de ficha <b>Summary</b>, encuentre y copie el ID.</p> <p><b>NOTA</b></p> <p>El sistema se conecta preferentemente al balanceador de carga basándose en el campo <b>kubernetes.io/elb.id</b>. Si no se especifica este campo, se utiliza el campo <b>spec.loadBalancerIP</b> (opcional y disponible solo en 1.23 y versiones anteriores).</p> <p>No utilice el campo <b>spec.loadBalancerIP</b> para conectarse al balanceador de carga. Este campo será descartado por Kubernetes. Para obtener más información, consulte <a href="#">Deprecation</a>.</p>	v1.9 o posterior
kubernetes.io/elb.autocreate	<a href="#">Tabla 7-17</a>	<p>Obligatorio <b>cuando los balanceadores de carga se crean automáticamente</b>.</p> <p><b>Ejemplo:</b></p> <ul style="list-style-type: none"> <li>● Si se creará automáticamente un balanceador de carga de red pública, establezca este parámetro en el siguiente valor:  <pre>{"type": "public", "bandwidth_name": "ce-bandwidth-1551163379627", "bandwidth_chargemode": "bandwidth", "bandwidth_size": 5, "bandwidth_sharetype": "PER", "eip_type": "5_bgp", "name": "james"}</pre> </li> <li>● Si se creará automáticamente un balanceador de carga de red privada, establezca este parámetro en el siguiente valor:  <pre>{"type": "inner", "name": "A-location-d-test"}</pre> </li> </ul>	v1.9 o posterior

Parámetro	Tipo	Descripción	Versión de clúster admitida
kubernetes.io/elb.enterpriseID	String	<p>Opcional <b>cuando los balanceadores de carga se crean automáticamente.</b></p> <p><b>Los clústeres de v1.15 y versiones posteriores admiten este campo. En clústeres anteriores a v1.15, los balanceadores de carga se crean en el proyecto predeterminado de forma predeterminada.</b></p> <p>Este parámetro indica el ID del proyecto de empresa en el que se creará el balanceador de carga de ELB.</p> <p>Si este parámetro no se especifica o se establece en <b>0</b>, los recursos estarán enlazados al proyecto de empresa predeterminado.</p> <p><b>Cómo obtenerlo:</b></p> <p>Inicie sesión en la consola de gestión y seleccione <b>Enterprise &gt; Project Management</b> en la barra de menú superior. En la lista que se muestra, haga clic en el nombre del proyecto de empresa de destino y copie el ID en la página de detalles del proyecto de empresa.</p>	v1.15 o posterior
kubernetes.io/elb.subnet-id	String	<p>Opcional <b>cuando los balanceadores de carga se crean automáticamente.</b></p> <p>ID de la subred donde se encuentra el clúster. El valor puede contener de 1 a 100 caracteres.</p> <ul style="list-style-type: none"> <li>● Obligatorio cuando se va a crear automáticamente un clúster de v1.11.7-r0 o anterior.</li> <li>● Opcional para los clústeres posteriores a v1.11.7-r0.</li> </ul>	<p>Obligatorio para las versiones anteriores a la v1.11.7-r0</p> <p>Descartado en las versiones posteriores a la v1.11.7-r0</p>

Parámetro	Tipo	Descripción	Versión de clúster admitida
kubernetes.io/elb.lb-algorithm	String	Especifica el algoritmo de equilibrio de carga del grupo de servidores backend. El valor predeterminado es <b>ROUND_ROBIN</b> . Opciones: <ul style="list-style-type: none"> <li>● <b>ROUND_ROBIN</b>: algoritmo de round robin ponderado</li> <li>● <b>LEAST_CONNECTIONS</b>: algoritmo de conexiones mínimas ponderadas</li> <li>● <b>SOURCE_IP</b>: algoritmo de hash IP de origen</li> </ul> NOTA Si este parámetro se establece en <b>SOURCE_IP</b> , la configuración de ponderación (campo <b>weight</b> ) de los servidores backend vinculados al grupo de servidores backend no es válida y no se puede habilitar la sesión adhesiva.	v1.9 o posterior

A continuación se muestra cómo utilizar las anotaciones anteriores:

- Asociar un balanceador de carga existente. Para obtener más información, véase [Uso de kubectl para crear un Service \(Uso de un balanceador de carga existente\)](#).

```

apiVersion: v1
kind: Service
metadata:
  name: nginx
  annotations:
    kubernetes.io/elb.id: <your_elb_id> # ELB ID.
    Replace it with the actual value.
    kubernetes.io/elb.class: performance # Load
balancer type
  kubernetes.io/elb.lb-algorithm: ROUND_ROBIN # Load
balancer algorithm
spec:
  selector:
    app: nginx
  ports:
  - name: service0
    port: 80
    protocol: TCP
    targetPort: 80
  type: LoadBalancer
    
```

- Creación automática de un balanceador de carga. Para obtener más información, véase [Uso de kubectl para crear un Service \(creación automática de un balanceador de carga\)](#).

Balanceador de carga compartido:

```

apiVersion: v1
kind: Service
metadata:
  annotations:
    kubernetes.io/elb.class: union
    
```

```

kubernetes.io/elb.autocreate: '{
  "type": "public",
  "bandwidth_name": "cce-bandwidth-1551163379627",
  "bandwidth_chargemode": "bandwidth",
  "bandwidth_size": 5,
  "bandwidth_sharetype": "PER",
  "eip_type": "5_bgp"
}'
  kubernetes.io/elb.enterpriseID: 0                # ID of the enterprise
project to which the load balancer belongs
  kubernetes.io/elb.lb-algorithm: ROUND_ROBIN     # Load balancer algorithm
labels:
  app: nginx
  name: nginx
spec:
  ports:
  - name: service0
    port: 80
    protocol: TCP
    targetPort: 80
  selector:
    app: nginx
  type: LoadBalancer
    
```

### Balanceador de carga dedicado:

```

apiVersion: v1
kind: Service
metadata:
  name: nginx
  labels:
    app: nginx
  namespace: default
  annotations:
    kubernetes.io/elb.class: performance
    kubernetes.io/elb.autocreate: '{
      "type": "public",
      "bandwidth_name": "cce-bandwidth-1626694478577",
      "bandwidth_chargemode": "bandwidth",
      "bandwidth_size": 5,
      "bandwidth_sharetype": "PER",
      "eip_type": "5_bgp",
      "available_zone": [
        ""
      ],
      "l4_flavor_name": "L4_flavor.elb.s1.small"
    }'
  kubernetes.io/elb.enterpriseID: 0                # ID of the enterprise
project to which the load balancer belongs
  kubernetes.io/elb.lb-algorithm: ROUND_ROBIN     # Load balancer algorithm
spec:
  selector:
    app: nginx
  ports:
  - name: cce-service-0
    targetPort: 80
    nodePort: 0
    port: 80
    protocol: TCP
  type: LoadBalancer
    
```

## Sesión adhesiva

**Tabla 7-10** Anotaciones para sesión adhesiva

Parámetro	Tipo	Descripción	Versión de clúster admitida
kubernetes.io/elb.session-affinity-mode	String	<p>Se admite la sesión adhesiva basada en la dirección IP de origen. Es decir, las solicitudes de acceso desde la misma dirección IP se reenvían al mismo servidor backend.</p> <ul style="list-style-type: none"> <li>● Deshabilitar sesión adhesiva: No configure este parámetro.</li> <li>● Activar la sesión adhesiva: Establezca este parámetro en <b>SOURCE_IP</b> para indicar que la sesión adhesiva se basa en la dirección IP de origen.</li> </ul> <p><b>NOTA</b>                      Cuando <b>kubernetes.io/elb.lb-algorithm</b> se establece en <b>SOURCE_IP</b> (algoritmo de dirección IP de origen), no se puede activar la sesión adhesiva.</p>	v1.9 o posterior
kubernetes.io/elb.session-affinity-option	<a href="#">Tabla 7-20</a>	Tiempo de espera de sesión adhesiva.	v1.9 o posterior

A continuación se muestra cómo utilizar las anotaciones anteriores:

```

apiVersion: v1
kind: Service
metadata:
  name: nginx
  annotations:
    kubernetes.io/elb.id: <your_elb_id> # ELB ID. Replace
    it with the actual value.
    kubernetes.io/elb.class: performance # Load balancer type
    kubernetes.io/elb.session-affinity-mode: SOURCE_IP # The sticky
    session type is source IP address.
    kubernetes.io/elb.session-affinity-option: '{"persistence_timeout":
    "30"}' # Stickiness duration (min)
spec:
  selector:
    app: nginx
  ports:
  - name: service0
    port: 80
    protocol: TCP
    targetPort: 80
  type: LoadBalancer
    
```

## Comprobación de estado

**Tabla 7-11** Anotaciones para la comprobación de estado

Parámetro	Tipo	Descripción	Versión de clúster admitida
kubernetes.io/elb.health-check-flag	String	Si se debe habilitar la comprobación de estado del ELB. <ul style="list-style-type: none"> <li>Habilitar la comprobación de estado: Deje en blanco este parámetro o configúrelo en <b>on</b>.</li> <li>Deshabilitar la comprobación de estado: Establezca este parámetro en <b>off</b>.</li> </ul> Si este parámetro está habilitado, el campo <a href="#">kubernetes.io/elb.health-check-option</a> también debe especificarse al mismo tiempo.	v1.9 o posterior
kubernetes.io/elb.health-check-option	<a href="#">Tabla 7-18</a>	Elementos de configuración de comprobación de estado de ELB.	v1.9 o posterior
kubernetes.io/elb.health-check-options	<a href="#">Tabla 7-19</a>	Concepto de configuración de comprobación de estado de ELB. Cada puerto de Service se puede configurar por separado y solo puede configurar algunos puertos. NOTA <a href="#">kubernetes.io/elb.health-check-option</a> y <a href="#">kubernetes.io/elb.health-check-options</a> no se pueden configurar al mismo tiempo.	v1.19.16-r5 o posterior v1.21.8-r0 o posterior v1.23.6-r0 o posterior v1.25.2-r0 o posterior

- A continuación, se muestra cómo utilizar [kubernetes.io/elb.health-check-option](#):

```

apiVersion: v1
kind: Service
metadata:
  name: nginx
  annotations:
    kubernetes.io/elb.id: <your_elb_id> # ELB ID.
    Replace it with the actual value.
    kubernetes.io/elb.class: performance # Load balancer
    
```

```

type
  kubernetes.io/elb.health-check-flag: 'on' # Enable the
ELB health check function.
  kubernetes.io/elb.health-check-option: '{
    "protocol": "TCP",
    "delay": "5",
    "timeout": "10",
    "max_retries": "3"
  }'
spec:
  selector:
    app: nginx
  ports:
  - name: service0
    port: 80
    protocol: TCP
    targetPort: 80
  type: LoadBalancer
    
```

- Para obtener más información sobre cómo usar `kubernetes.io/elb.health-check-options` en [Configuración de la comprobación de estado para varios puertos](#).

## Protocolo HTTP

Tabla 7-12 Anotaciones para el uso de protocolos HTTP

Parámetro	Tipo	Descripción	Versión de clúster admitida
kubernetes.io/elb.protocol-port	String	Puerto de configuración de reenvío de capa 7 utilizado por el Service.	v1.19.16 o posterior
kubernetes.io/elb.cert-id	String	Certificado HTTP utilizado por el Service para el reenvío de capa 7.	v1.19.16 o posterior

Para obtener más información sobre los escenarios de la aplicación, consulte [Service usando HTTP](#).

## Ajuste dinámico de ponderación del backend ECS

**Tabla 7-13** Anotaciones para ajustar dinámicamente la ponderación del ECS backend

Parámetro	Tipo	Descripción	Versión de clúster admitida
kubernetes.io/elb.adaptive-weight	String	<p>Ajusta dinámicamente la ponderación del ECS backend del balanceador de carga basado en pods. Las solicitudes recibidas por cada pod son más equilibradas.</p> <ul style="list-style-type: none"> <li>● <b>true</b>: activado</li> <li>● <b>false</b>: desactivado</li> </ul> <p>Este parámetro solo se aplica a clústeres de v1.21 o posterior y no es válido en redes de passthrough.</p>	v1.21 o posterior

A continuación se muestra cómo utilizar las anotaciones anteriores:

```

apiVersion: v1
kind: Service
metadata:
  name: nginx
  annotations:
    kubernetes.io/elb.id: <your_elb_id> # ELB ID. Replace
    kubernetes.io/elb.class: performance # Load balancer type
    kubernetes.io/elb.adaptive-weight: 'true' # Enable dynamic
    adjustment of the weight of the backend ECS.
spec:
  selector:
    app: nginx
  ports:
  - name: service0
    port: 80
    protocol: TCP
    targetPort: 80
  type: LoadBalancer
    
```

## Capacidad de pass-through

**Tabla 7-14** Anotaciones para la capacidad de pass-through

Parámetro	Tipo	Descripción	Versión de clúster admitida
kubernetes.io/elb.pass-through	String	Si las solicitudes de acceso desde el clúster al Service pasan a través del balanceador de carga de ELB.	v1.19 o posterior

Para obtener más información sobre los escenarios de la aplicación, consulte [Habilitación de redes de paso a través para los servicios de LoadBalancer](#).

## Lista blanca

**Tabla 7-15** Anotaciones para la lista de acceso de ELB

Parámetro	Tipo	Descripción	Versión de clúster admitida
kubernetes.io/elb.acl-id	String	<p>Este parámetro es obligatorio cuando se establece una lista blanca de direcciones IP para un balanceador de carga. El valor de este parámetro es el ID de grupo de direcciones IP del balanceador de carga. Para obtener más información, consulte <a href="#">Grupo de direcciones IP</a>.</p> <p><b>Este parámetro solo tiene efecto para los balanceadores de carga dedicados y solo tiene efecto cuando se crea un Service o se especifica un nuevo puerto de servicio (oyente).</b></p> <p><b>Cómo obtenerlo:</b></p> <p>Inicie sesión en la consola. En <b>Service List</b>, seleccione <b>Networking &gt; Elastic Load Balance</b>. En la Consola de red, elija <b>Elastic Load Balance &gt; IP Address Groups</b> y copie el <b>ID</b> del grupo de direcciones IP de destino.</p>	v1.19.1 6 v1.21.4
kubernetes.io/elb.acl-status	String	<p>Este parámetro es obligatorio cuando se establece una lista blanca de direcciones IP para un balanceador de carga. El valor es <b>on</b>, que indica que el control de acceso está habilitado.</p> <p><b>Este parámetro solo tiene efecto para los balanceadores de carga dedicados y solo tiene efecto cuando se crea un Service o se especifica un nuevo puerto de servicio (oyente).</b></p>	v1.19.1 6 v1.21.4

Parámetro	Tipo	Descripción	Versión de clúster admitida
kubernetes.io/elb.acl-type	String	<p>Este parámetro es obligatorio cuando se establece la lista blanca de direcciones IP para un balanceador de carga.</p> <ul style="list-style-type: none"> <li>● <b>black</b>: indica la lista negra. El grupo de direcciones IP seleccionado no puede acceder a la dirección de ELB.</li> <li>● <b>white</b>: indica la lista blanca. Solo el grupo de direcciones IP seleccionado puede acceder a la dirección de ELB.</li> </ul> <p><b>Este parámetro solo tiene efecto para los balanceadores de carga dedicados y solo tiene efecto cuando se crea un Service o se especifica un nuevo puerto de servicio (oyente).</b></p>	v1.19.16 v1.21.4

A continuación se muestra cómo utilizar las anotaciones anteriores:

```

apiVersion: v1
kind: Service
metadata:
  name: nginx
  annotations:
    kubernetes.io/elb.id: <your_elb_id> # ELB ID. Replace it
    kubernetes.io/elb.class: performance # Load balancer type
    kubernetes.io/elb.acl-id: <your_acl_id> # ELB IP address group
    kubernetes.io/elb.acl-status: 'on' # Enable access control.
    kubernetes.io/elb.acl-type: 'white' # Whitelist control
spec:
  selector:
    app: nginx
  ports:
    - name: service0
      port: 80
      protocol: TCP
      targetPort: 80
  type: LoadBalancer
    
```

## Red de host

**Tabla 7-16** Anotaciones para la red host

Parámetro	Tipo	Descripción	Versión de clúster admitida
kubernetes.io/hws-hostNetwork	String	Si el pod usa <b>hostNetwork</b> , el ELB reenvía la solicitud a la red anfitriona después de usar esta anotación.  Opciones: <ul style="list-style-type: none"> <li>● <b>true</b>: activado</li> <li>● <b>false</b> (predeterminado): desactivado</li> </ul>	v1.9 o posterior

A continuación se muestra cómo utilizar las anotaciones anteriores:

```

apiVersion: v1
kind: Service
metadata:
  name: nginx
  annotations:
    kubernetes.io/elb.id: <your_elb_id> # ELB ID. Replace
it with the actual value.
    kubernetes.io/elb.class: performance # Load balancer type
    kubernetes.io/hws-hostNetwork: 'true' # The load balancer
forwards the request to the host network.
spec:
  selector:
    app: nginx
  ports:
  - name: service0
    port: 80
    protocol: TCP
    targetPort: 80
  type: LoadBalancer
    
```

## Estructura de datos

**Tabla 7-17** Estructura de datos del campo **elb.autocreate**

Parámetro	Obligatorio	Tipo	Descripción
name	No	String	Nombre del balanceador de carga creado automáticamente. Rango de valores: de 1 a 64 caracteres, incluidas las letras minúsculas, los dígitos y los guiones bajos (_). El valor debe comenzar con una letra minúscula y terminar con una letra minúscula o un dígito. Predeterminado: <b>cce-lb+service.UID</b>
type	No	String	Tipo de red del balanceador de carga. <ul style="list-style-type: none"> <li>● <b>public</b>: balanceador de carga de red pública</li> <li>● <b>inner</b>: balanceador de carga de red privada</li> </ul> Predeterminado: <b>inner</b>
bandwidth_name	Sí para los balanceadores de carga de red pública	String	Nombre del ancho de banda. El valor predeterminado es <b>cce-bandwidth-*****</b> . Rango de valores: de 1 a 64 caracteres, incluidas las letras minúsculas, los dígitos y los guiones bajos (_). El valor debe comenzar con una letra minúscula y terminar con una letra minúscula o un dígito.
bandwidth_chargemode	No	String	Modo de facturación de ancho de banda. <ul style="list-style-type: none"> <li>● <b>bandwidth</b>: facturado por ancho de banda</li> <li>● <b>traffic</b>: facturado por tráfico</li> </ul> Predeterminado: <b>bandwidth</b>
bandwidth_size	Sí para los balanceadores de carga de red pública	Integer	Tamaño del ancho de banda. El valor predeterminado es de 1 a 2000 Mbit/s. Configure este parámetro en función del rango de ancho de banda permitido en su región.

Parámetro	Obligatorio	Tipo	Descripción
bandwidth_sharetype	Sí para los balanceadores de carga de red pública	String	Modo de uso compartido de ancho de banda. <ul style="list-style-type: none"> <li>● <b>PER</b>: Ancho de banda dedicado</li> </ul>
eip_type	Sí para los balanceadores de carga de red pública	String	Tipo de la EIP. <ul style="list-style-type: none"> <li>● <b>5_telcom</b>: China Telecom</li> <li>● <b>5_union</b>: China Unicom</li> <li>● <b>5_bgp</b>: BGP dinámico</li> <li>● <b>5_sbgp</b>: BGP estático</li> </ul>
vip_subnet_cidr_id	No	String	Subred donde se encuentra el balanceador de carga. Este campo es compatible con clústeres de v1.21 o posterior. Si no se especifica este parámetro, el balanceador de carga de ELB y el clúster están en la misma subred.
available_zone	Sí	Array of strings	AZ donde se encuentra el balanceador de carga. Puede obtener todas las AZ soportadas por <a href="#">consultar la lista de AZ</a> . Este parámetro solo está disponible para los balanceadores de carga dedicados.
l4_flavor_name	Sí	String	Nombre de la variante del balanceador de carga de capa 4. Puede obtener todos los tipos admitidos <a href="#">consultando la lista de variantes</a> . Este parámetro solo está disponible para los balanceadores de carga dedicados.
l7_flavor_name	No	String	Nombre de la variante del balanceador de carga de capa 7. Puede obtener todos los tipos admitidos <a href="#">consultando la lista de variantes</a> . Este parámetro solo está disponible para los balanceadores de carga dedicados. El valor de este parámetro debe ser el mismo que el de <b>l4_flavor_name</b> , es decir, ambas son especificaciones elásticas o especificaciones fijas.

Parámetro	Obligatorio	Tipo	Descripción
elb_virsubnet_ids	No	Array of strings	<p>Subred donde se encuentra el servidor de backend del balanceador de carga. Si este parámetro se deja en blanco, se utiliza la subred de clúster predeterminada. Los balanceadores de carga ocupan un número diferente de direcciones IP de subred según sus especificaciones. Por lo tanto, no se recomienda utilizar los bloques CIDR de subred de otros recursos (como clústeres y nodos) como el bloque CIDR del balanceador de carga.</p> <p>Este parámetro solo está disponible para los balanceadores de carga dedicados.</p> <p>Por ejemplo:</p> <pre>"elb_virsubnet_ids": [   "14567f27-8ae4-42b8-ae47-9f847a4690dd" ]</pre>

**Tabla 7-18** Descripción de la estructura de datos del campo **elb.health-check-option**

Parámetro	Obligatorio	Tipo	Descripción
delay	No	String	<p>Tiempo de espera inicial (en segundos) para iniciar la comprobación de estado.</p> <p>Rango de valores: 1 a 50. Valor predeterminado: <b>5</b></p>
timeout	No	String	<p>Tiempo de espera de la comprobación de estado, en segundos.</p> <p>Rango de valores: 1 a 50. Valor predeterminado: <b>10</b></p>
max_retries	No	String	<p>Número máximo de reintentos de comprobación de estado.</p> <p>Rango de valores: 1 a 10. Valor predeterminado: <b>3</b></p>
protocol	No	String	<p>Protocolo de comprobación de estado.</p> <p>Opciones de valor: TCP o HTTP</p>

Parámetro	Obligatorio	Tipo	Descripción
path	No	String	URL de comprobación de estado. Este parámetro debe configurarse cuando el protocolo es HTTP. Valor predeterminado: / El valor puede contener de 1 a 10,000 caracteres.

**Tabla 7-19** Descripción de la estructura de datos del campo **elb.health-check-options**

Parámetro	Obligatorio	Tipo	Descripción
target_service_port	Sí	String	Puerto para comprobación de estado especificado por spec.ports. El valor consiste en el protocolo y el número de puerto, por ejemplo, TCP:80.
monitor_port	No	String	Puerto reespecificado para la comprobación de estado. Si no se especifica este parámetro, el puerto de servicio se utiliza de forma predeterminada. <b>NOTA</b> Asegúrese de que el puerto está en el estado de escucha en el nodo donde se encuentra el pod. De lo contrario, el resultado de la comprobación de estado se verá afectado.
delay	No	String	Tiempo de espera inicial (en segundos) para iniciar la comprobación de estado. Rango de valores: 1 a 50. Valor predeterminado: <b>5</b>
timeout	No	String	Tiempo de espera de la comprobación de estado, en segundos. Rango de valores: 1 a 50. Valor predeterminado: <b>10</b>
max_retries	No	String	Número máximo de reintentos de comprobación de estado. Rango de valores: 1 a 10. Valor predeterminado: <b>3</b>
protocol	No	String	Protocolo de comprobación de estado. Valor predeterminado: protocolo del Service asociado Opción de valor: TCP, UDP o HTTP

Parámetro	Obligatorio	Tipo	Descripción
path	No	String	URL de comprobación de estado. Este parámetro debe configurarse cuando el protocolo es HTTP. Valor predeterminado: / El valor puede contener de 1 a 10,000 caracteres.

**Tabla 7-20** Estructura de datos del campo `elb.session-affinity-option`

Parámetro	Obligatorio	Tipo	Descripción
persistence_timeout	Sí	String	Tiempo de espera de sesión adhesiva, en minutos. Este parámetro solo es válido cuando <code>elb.session-affinity-mode</code> está establecido en <code>SOURCE_IP</code> . Rango de valores: 1 a 60. Valor predeterminado: <b>60</b>

### 7.3.4.3 Service usando HTTP

#### Restricciones

- Solo los clústeres de v1.19.16 o posterior admiten HTTP.
- No conecte el ingreso y el Service que utiliza HTTP al mismo oyente del mismo balanceador de carga. De lo contrario, se produce un conflicto de puerto.
- El enrutamiento de capa 7 de ELB se puede habilitar para los Services. Tanto los balanceadores de carga de ELB compartidos como los dedicados pueden estar interconectados.

Las restricciones en los balanceadores de carga de ELB dedicados son las siguientes:

- Para interconectarse con un balanceador de carga dedicado existente, la variante de balanceador de carga **debe soportar tanto el enrutamiento de capa 4 como de capa 7**. De lo contrario, el balanceador de carga no funcionará como se espera.
- Si utiliza un balanceador de carga creado automáticamente, no puede utilizar la consola de CCE para crear automáticamente un balanceador de carga dedicado de capa 7. En su lugar, puede usar YAML para crear un balanceador de carga dedicado de capa 7, usar las capacidades de capa 4 y capa 7 de la instancia de ELB exclusiva (es decir, especifique las variantes de capa 4 y capa 7 en la anotación de `kubernetes.io/elb.autocreate`).

#### Service usando HTTP

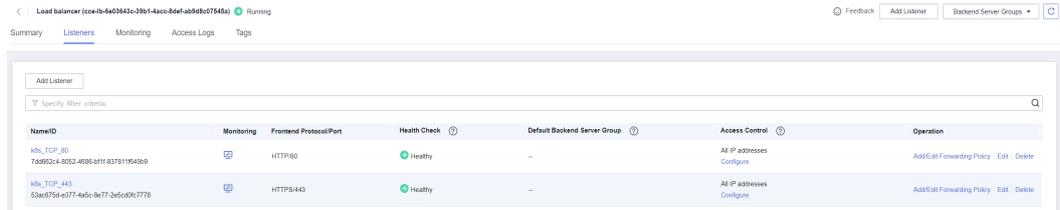
Es necesario agregar las siguientes anotaciones:

- **kubernetes.io/elb.protocol-port:** "https:443,http:80"  
El valor de **protocol-port** debe ser el mismo que el puerto en el campo **spec.ports** del Service. El formato es *Protocol:Port*. El puerto coincide con el del campo **service.spec.ports** y se libera como el protocolo correspondiente.
- **kubernetes.io/elb.cert-id:** "17e3b4f4bc40471c86741dc3aa211379"  
**cert-id** indica el ID de certificado en la gestión de certificados de ELB. Cuando **https** está configurado para **protocol-port**, el certificado del oyente de ELB se establecerá en el certificado **cert-id**. Cuando se liberan varios servicios de HTTPS, se utiliza el mismo certificado.

El siguiente es un ejemplo de configuración. Los dos puertos de **spec.ports** corresponden a los de **kubernetes.io/elb.protocol-port**. Los puertos 443 y 80 están habilitados para solicitudes de HTTPS y de HTTP, respectivamente.

```
apiVersion: v1
kind: Service
metadata:
  annotations:
    # When an ELB load balancer is automatically created, both layer-4 and
    layer-7 flavors need to be specified.
    kubernetes.io/elb.autocreate: '
      {
        "type": "public",
        "bandwidth_name": "cce-bandwidth-1634816602057",
        "bandwidth_chargemode": "bandwidth",
        "bandwidth_size": 5,
        "bandwidth_sharetype": "PER",
        "eip_type": "5_bgp",
        "available_zone": [
          ""
        ],
        "l7_flavor_name": "L7_flavor.elb.s2.small",
        "l4_flavor_name": "L4_flavor.elb.s1.medium"
      }'
    kubernetes.io/elb.class: performance
    kubernetes.io/elb.protocol-port: "https:443,http:80"
    kubernetes.io/elb.cert-id: "17e3b4f4bc40471c86741dc3aa211379"
  labels:
    app: nginx
    name: test
  name: test
  namespace: default
spec:
  ports:
    - name: cce-service-0
      port: 443
      protocol: TCP
      targetPort: 80
    - name: cce-service-1
      port: 80
      protocol: TCP
      targetPort: 80
  selector:
    app: nginx
    version: v1
  sessionAffinity: None
  type: LoadBalancer
```

Utilice las configuraciones de ejemplo anteriores para crear un Service. En el nuevo balanceador de carga de ELB, puede ver que se crean los oyentes en los puertos 443 y 80.



### 7.3.4.4 Configuración de la comprobación de estado para varios puertos

El campo de anotación relacionado con la comprobación de estado del LoadBalancer Service se actualiza de **Kubernetes.io/elb.health-check-option** a **Kubernetes.io/elb.health-check-options**. Cada puerto de Service se puede configurar por separado y solo puede configurar algunos puertos. Si el protocolo de puerto no necesita configurarse por separado, el campo de anotación original todavía está disponible y no necesita modificarse.

### Restricciones

- Esta función solo tiene efecto en las siguientes versiones:
  - v1.19: v1.19.16-r5 o posterior
  - v1.21: v1.21.8-r0 o posterior
  - v1.23: v1.23.6-r0 o posterior
  - v1.25: v1.25.2-r0 o posterior
- **kubernetes.io/elb.health-check-option** y **kubernetes.io/elb.health-check-options** no se pueden configurar al mismo tiempo.
- El campo **target\_service\_port** es obligatorio y debe ser único.
- Para un puerto TCP, el protocolo de comprobación de estado solo puede ser TCP o HTTP. Para un puerto UDP, el protocolo de comprobación de estado debe ser UDP.

### Procedimiento

A continuación se muestra un ejemplo de uso de la anotación **kubernetes.io/elb.health-check-options**:

```
apiVersion: v1
kind: Service
metadata:
  name: nginx
  namespace: default
  labels: {}
  annotations:
    kubernetes.io/elb.class: union
    kubernetes.io/elb.id: 038ffbda-bd3a-48bb-8b8c-a8582601fd97
    kubernetes.io/elb.lb-algorithm: ROUND_ROBIN
    kubernetes.io/elb.health-check-flag: 'on'
    kubernetes.io/elb.health-check-options: '{
      "target_service_port": "TCP:80", // (Mandatory) Port for health check
      specified by spec.ports. The value consists of the protocol and port number, for
      example, TCP:80.
      "monitor_port": "", // (Optional) Re-specified port for health check.
      If this parameter is not specified, the service port is used by default. Ensure
      that the port is in the listening state on the node where the pod is located.
      Otherwise, the health check result will be affected.
      "protocol":"TCP",
      "delay":"5",
      "timeout":"10",
      "max_retries":"3",
      "path":"/"
    }'
```

```

    }'
spec:
  selector: {}
  externalTrafficPolicy: Cluster
  ports:
  - name: cce-service-0
    targetPort: 80
    nodePort: 0
    port: 80
    protocol: TCP
  type: LoadBalancer
  loadBalancerIP: **.*.*.*.*
    
```

**Tabla 7-21** Descripción de la estructura de datos del campo **elb.health-check-options**

Parámetro	Obligatorio	Tipo	Descripción
target_service_port	Sí	String	Puerto para comprobación de estado especificado por spec.ports. El valor consiste en el protocolo y el número de puerto, por ejemplo, TCP:80.
monitor_port	No	String	Puerto reespecificado para la comprobación de estado. Si no se especifica este parámetro, el puerto de servicio se utiliza de forma predeterminada. <b>NOTA</b> Asegúrese de que el puerto está en el estado de escucha en el nodo donde se encuentra el pod. De lo contrario, el resultado de la comprobación de estado se verá afectado.
delay	No	String	Tiempo de espera inicial (en segundos) para iniciar la comprobación de estado. Rango de valores: 1 a 50. Valor predeterminado: <b>5</b>
timeout	No	String	Tiempo de espera de la comprobación de estado, en segundos. Rango de valores: 1 a 50. Valor predeterminado: <b>10</b>
max_retries	No	String	Número máximo de reintentos de comprobación de estado. Rango de valores: 1 a 10. Valor predeterminado: <b>3</b>
protocol	No	String	Protocolo de comprobación de estado. Valor predeterminado: protocolo del Service asociado Opción de valor: TCP, UDP o HTTP

Parámetro	Obligatorio	Tipo	Descripción
path	No	String	URL de comprobación de estado. Este parámetro debe configurarse cuando el protocolo es HTTP. Valor predeterminado: / El valor puede contener de 1 a 10,000 caracteres.

### 7.3.4.5 Configuración del estado del pod a través de la comprobación de estado de ELB

El estado listo del pod está asociado con la comprobación del estado de ELB. Después de que la comprobación de estado es exitosa, el pod está listo. Esta asociación funciona con los parámetros **strategy.rollingUpdate.maxSurge** y **strategy.rollingUpdate.maxUnavailable** del pod para implementar una actualización gradual elegante.

#### Restricciones

- Esta función solo tiene efecto en las siguientes versiones:
  - v1.19: v1.19.16-r5 o posterior
  - v1.21: v1.21.8-r0 o posterior
  - v1.23: v1.23.6-r0 o posterior
  - v1.25: v1.25.2-r0 o posterior
- Esta función solo se aplica a escenarios de transferencia, es decir, escenarios en los que se utilizan balanceadores de carga dedicados en clústeres de CCE Turbo.
- Para usar esta función, debe configurar el campo **readinessGates** en el pod y especificar la etiqueta **target-health.elb.k8s.cce/{serviceName}** donde **{serviceName}** indica el nombre del servicio.
- El estado listo para el pod solo tiene efecto cuando el backend ELB está conectado inicialmente. El estado de comprobación de estado posterior no afecta al estado listo para el pod.

### Configuración del estado del pod a través de la comprobación de estado de ELB

Para utilizar Pod readiness Gates, lleve a cabo los siguientes pasos:

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

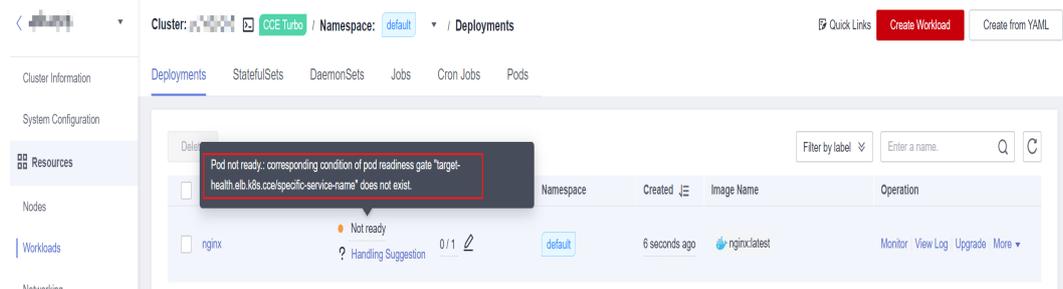
**Paso 2** En el panel de navegación, elija **Workloads**. En la esquina superior derecha, haga clic en **Create from YAML**.

Contenido de YAML:

```
kind: Deployment
apiVersion: apps/v1
metadata:
  name: nginx
  namespace: default
  labels:
    version: v1
```

```
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
      version: v1
  template:
    metadata:
      labels:
        app: nginx
        version: v1
    spec:
      containers:
        - name: container-1
          image: nginx:latest
      imagePullSecrets:
        - name: default-secret
      readinessGates:
        - conditionType: target-health.elb.k8s.cce/specific-service-name #
Specifies the ServiceName.
      strategy:
        type: RollingUpdate
      rollingUpdate:
        maxUnavailable: 25% # Works with the following two parameters to
control the number of ELB backends and implement graceful rolling upgrade.
        maxSurge: 25%
```

**Paso 3** Haga clic en **OK**. En la lista de carga de trabajo, puede comprobar el estado de la carga de trabajo y encontrar que el pod no está listo.



**Paso 4** En el panel de navegación, elija **Networking**. En la esquina superior derecha, haga clic en **Create Service** y establezca los siguientes parámetros:

- **Service Name:** El valor debe ser el mismo que el valor de **readinessGates** en el pod.
- **Service Type:** Seleccione **LoadBalancer**.
- **Selector:** haga clic en **Reference Workload Label**, seleccione la carga de trabajo creada en el paso anterior y haga clic en **OK**.
- **Load Balancer:** Se deben utilizar balanceadores de carga dedicados. Puede seleccionar un balanceador de carga existente o crear automáticamente un balanceador de carga.
- **Set ELB:** Habilite la comprobación de estado. (De lo contrario, la comprobación de estado se realiza correctamente por defecto.)

**Create Service**

Service Name:

Service Type: ClusterIP ClusterIP NodePort NodePort **LoadBalancer LoadBalancer** DNAT NatGateway

Service Affinity: **Cluster-level** Node-level ?

Namespace: **default**

Selector: Key = Value  [Reference Workload Label](#)  
 app = nginx ✓ version = v1 ✓  
Services are associated with workloads (labels) through selectors.

Load Balancer: **Dedicat...** Use exi... elb-turno-test  [Create Load Balancer](#)  
Supports only dedicated load balancers of Network type in VPC vpc-ipv6 where the cluster resides.  
 Set ELB: Load balancing algorithm: Weighted round robin; Sticky session: Disable; [↗](#)  
 I have read [Notes on Using Load Balancers](#).

Health Check: Disable **Global health check** Custom health check  
 protocol: TCP delay(s): 5 | timeout(s): 10 | maxRetries: 3 [↗](#)

**Paso 5** Vaya a la consola de ELB y compruebe el grupo de servidores backend. El estado de la comprobación de estado es normal.

**Paso 6** En la consola de CCE, la carga de trabajo está en el estado **Running**.

Deployments StatefulSets DaemonSets Jobs Cron Jobs Pods

Workload Name	Status	Pods (Normal/All)
nginx	<span style="color: green;">●</span> Running	1 / 1

----Fin

### 7.3.4.6 Habilidad de redes de paso a través para los servicios de LoadBalancer

#### Desafíos

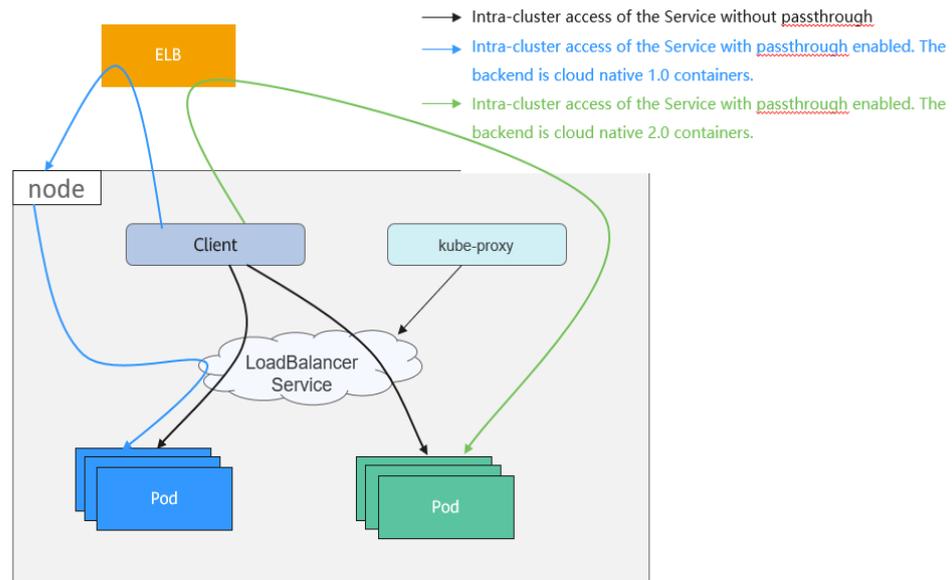
Un clúster de Kubernetes puede publicar aplicaciones que se ejecutan en un grupo de pods como Services, que proporcionan entradas de acceso de capa 4 unificadas. Para un Service de Loadbalancer, kube-proxy configura el LoadbalancerIP en **status** del Service a la regla de reenvío local del nodo de forma predeterminada. Cuando un pod accede al balanceador de carga desde dentro del clúster, el tráfico se reenvía dentro del clúster en lugar de ser reenviado por el balanceador de carga.

kube-proxy es responsable del reenvío dentro del clúster. kube-proxy tiene dos modos de reenvío: iptables e IPVS. iptables es un modo de reenvío de sondeo simple. IPVS tiene múltiples modos de reenvío, pero requiere modificar los parámetros de inicio de kube-proxy. En comparación con iptables e IPVS, los balanceadores de carga proporcionan políticas de reenvío más flexibles, así como capacidades de comprobación de estado.

## Solución

CCE admite la creación de redes de paso a través. Puede configurar el **annotation** de `kubernetes.io/elb.pass-through` para el Service Loadbalancer. El acceso dentro del clúster a la dirección del balanceador de carga de Service se reenvía entonces a los pods de backend por el balanceador de carga.

**Figura 7-25** Ilustración de red de paso



- Clústeres de CCE
 

Cuando se accede a un Service de LoadBalancer dentro del clúster, el acceso se reenvía a los pods de backend mediante iptables/IPVS de forma predeterminada.

Cuando se accede a un Service LoadBalancer (configurado con `elb.pass-through`) dentro del clúster, el acceso se reenvía primero al balanceador de carga, luego a los nodos y, finalmente, a los pods del backend usando iptables/IPVS.
- Clústeres de CCE Turbo
 

Cuando se accede a un Service de LoadBalancer dentro del clúster, el acceso se reenvía a los pods de backend mediante iptables/IPVS de forma predeterminada.

Cuando se accede a un Service de LoadBalancer (configurado con `elb.pass-through`) dentro del clúster, el acceso se reenvía primero al balanceador de carga y, a continuación, a los pods.

## Notas y restricciones

- Después de configurar las redes de paso a través para un balanceador de carga dedicado, no se puede acceder a contenedores en el nodo donde se ejecuta la carga de trabajo a través del Service.
- Las redes de paso a través no son compatibles con clústeres de v1.15 o anteriores.
- En el modo de red IPVS, la configuración de paso a través del Service conectado al mismo ELB debe ser la misma.

## Procedimiento

Esta sección describe cómo crear una Deployment usando una imagen de Nginx y crear un Service con redes de paso a través habilitadas.

### Paso 1 Utilice la imagen Nginx para crear una Deployment.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx:latest
        name: container-0
        resources:
          limits:
            cpu: 100m
            memory: 200Mi
          requests:
            cpu: 100m
            memory: 200Mi
      imagePullSecrets:
      - name: default-secret
```

### Paso 2 Cree un Service de LoadBalancer y configure `kubernetes.io/elb.pass-through` en `true`.

Para obtener más información sobre cómo crear un Service LoadBalancer, consulte [Creación automática de un Service LoadBalancer](#).

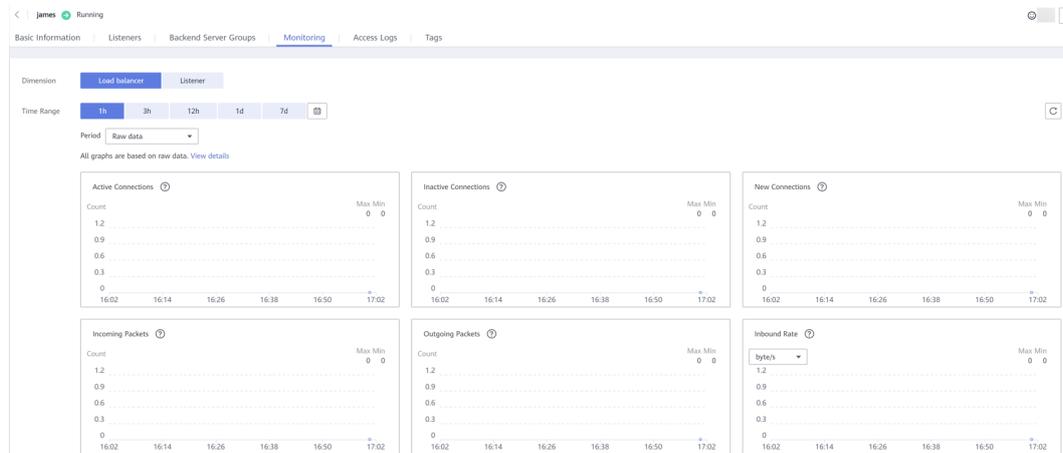
```
apiVersion: v1
kind: Service
metadata:
  annotations:
    kubernetes.io/elb.pass-through: "true"
    kubernetes.io/elb.class: union
    kubernetes.io/elb.autocreate: '{"type":"public","bandwidth_name":"cce-bandwidth","bandwidth_chargemode":"bandwidth","bandwidth_size":5,"bandwidth_sharetpe":"PER","eip_type":"5_bgp","name":"james"}'
  labels:
    app: nginx
  name: nginx
spec:
  externalTrafficPolicy: Local
  ports:
  - name: service0
    port: 80
    protocol: TCP
    targetPort: 80
  selector:
    app: nginx
  type: LoadBalancer
```

Se crea automáticamente un balanceador de carga compartido denominado **james**. Utilice `kubernetes.io/elb.subnet-id` para especificar la subred de VPC donde se encuentra el balanceador de carga. El balanceador de carga y el clúster deben estar en la misma VPC.

----Fin

## Verificación

Compruebe el balanceador de carga de ELB correspondiente al Service creado. El nombre del balanceador de carga es **james**. El número de conexiones de ELB es **0** como se muestra en la siguiente figura.



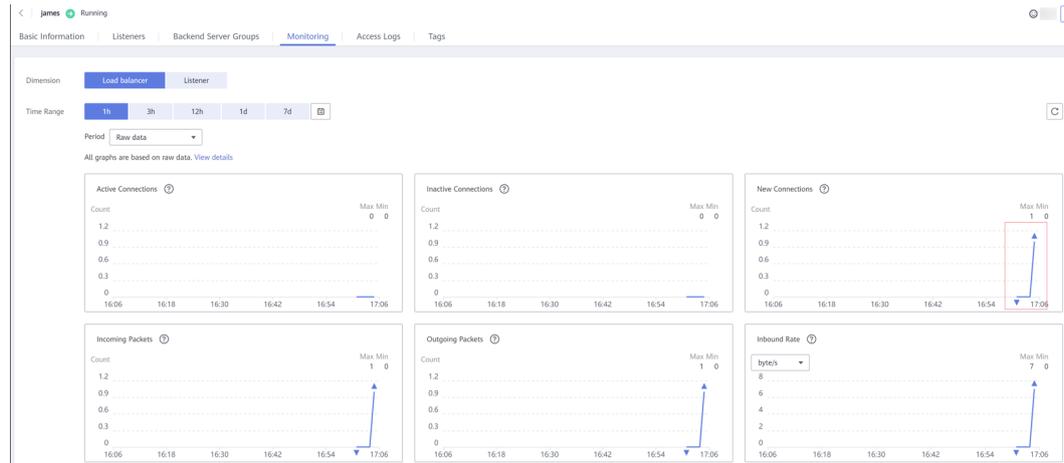
Utilice `kubectl` para conectarse al clúster, vaya a un contenedor de Nginx y acceda a la dirección de ELB. El acceso se realiza correctamente.

```
# kubectl get pod
NAME                                READY   STATUS    RESTARTS   AGE
nginx-7c4c5cc6b5-vpncx             1/1     Running   0           9m47s
nginx-7c4c5cc6b5-xj5wl             1/1     Running   0           9m47s
# kubectl exec -it nginx-7c4c5cc6b5-vpncx -- /bin/sh
# curl 120.46.141.192
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

Espera un período de tiempo y vea los datos de supervisión del ELB. Se crea una nueva conexión de acceso para el ELB, que indica que el acceso pasa a través del balanceador de carga del ELB como se esperaba.



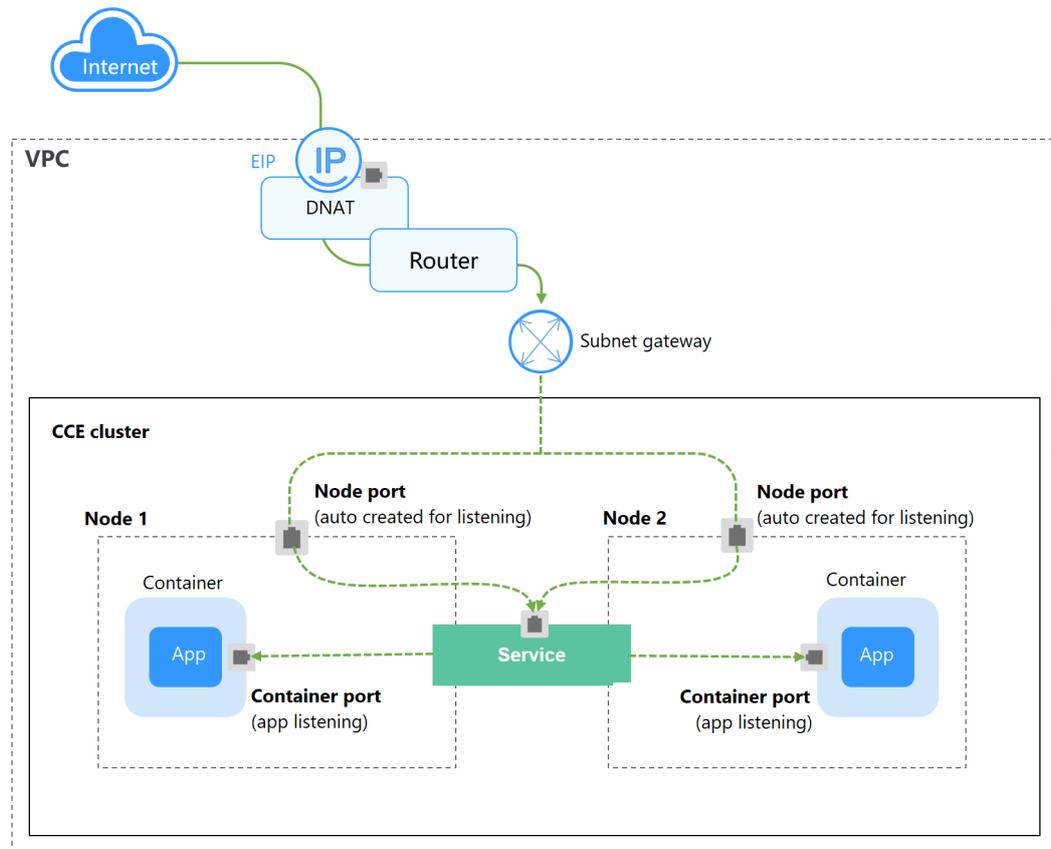
## 7.3.5 DNAT

### Escenario

Un **gateway de traducción de direcciones de red de destino (DNAT)** está situado entre los nodos de clúster y las redes públicas y se le asigna una EIP. Después de recibir solicitudes entrantes de redes públicas, el gateway de NAT traduce la EIP (dirección de destino en las solicitudes entrantes) en una dirección interna de clúster. A los usuarios de la carga de trabajo les parece que todos los nodos que ejecutan la carga de trabajo comparten la misma EIP.

DNAT proporciona mayor fiabilidad que el NodePort basado en EIP en el que la EIP está vinculada a un único nodo y una vez que el nodo está inactivo, todas las solicitudes entrantes a la carga de trabajo no se distribuirán. La dirección de acceso tiene el formato de <EIP>:<puerto de acceso>, por ejemplo, 10.117.117.117:80.

Figura 7-26 DNAT

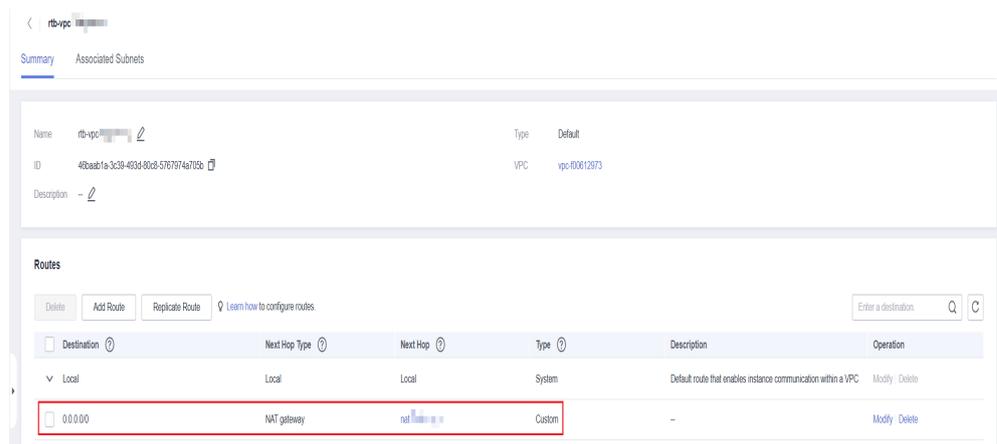


## Notas y restricciones

Tenga en cuenta las siguientes restricciones cuando utilice el servicio de NAT Gateway:

- Los clústeres que utilizan el modelo de red de VPC no permiten que contenedores acceda a los servicios de DNAT cuyo **externalTrafficPolicy** está establecido en **local**.
- Múltiples reglas para un gateway de NAT pueden usar la misma EIP, pero las reglas para diferentes gateway de NAT deben usar diferentes EIP.
- Cada VPC puede tener solo un gateway de NAT.
- Los usuarios no pueden agregar manualmente la ruta predeterminada en una VPC.
- Solo se puede agregar una regla de SNAT a una subred en una VPC.
- Las reglas de SNAT y de DNAT están diseñadas para diferentes funciones. Si las reglas de SNAT y de DNAT usan la misma EIP, se producirá una preferencia de recursos. Una regla de SNAT no puede compartir una EIP con una regla de DNAT con **Port Type** establecida en **All ports**.
- Las reglas de DNAT no admiten la vinculación de una EIP a una dirección IP virtual.
- Cuando se configuren los servicios EIP y NAT Gateway para un servidor, los datos serán reenviados a través de la EIP.
- El bloque CIDR personalizado debe ser un subconjunto de los bloques CIDR de subred de VPC.
- El bloque CIDR personalizado debe ser un bloque CIDR de Direct Connect y no puede entrar en conflicto con los bloques CIDR de subred existentes de VPC.

- Cuando realiza operaciones en recursos subyacentes de un ECS, por ejemplo, cambiando sus especificaciones, las reglas de gateway de NAT configuradas se vuelven inválidas. Es necesario eliminar las reglas y volver a configurarlas.
- Después de crear un Service, si la configuración de afinidad se cambia del nivel de clúster al nivel de nodo, la tabla de seguimiento de conexiones no se borrará. Se recomienda no modificar la configuración de afinidad del Service después de que se haya creado el Servicio. Si necesita modificarlo, vuelva a crear un Service.
- Si la subred de nodo está asociada con una tabla de ruta personalizada, debe agregar la ruta de NAT a la tabla de ruta personalizada cuando use el Service de DNAT.



## Creación de un gateway de NAT y una dirección IP elástica

Ha creado un gateway de NAT y una dirección IP elástica. El procedimiento específico es el siguiente:

- Paso 1** Inicie sesión en la consola de gestión, elija **Networking** > **NAT Gateway** en la lista de servicios y haga clic en **Buy NAT Gateway** en la esquina superior derecha. Configure los parámetros basados en los requisitos del sitio.

### 📖 NOTA

Al comprar un gateway de NAT, asegúrese de que el gateway de NAT pertenece a la misma VPC y subred que el clúster de CCE donde se ejecuta la carga de trabajo.

- Paso 2** Inicie sesión en la consola de gestión, elija **Networking** > **Elastic IP** en la lista de servicios y haga clic en **Buy EIP** en la esquina superior derecha. Configure los parámetros según los requisitos del sitio.

----Fin

## Creación de un servicio de gateway de DNAT

- Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.
- Paso 2** Elija **Networking** en el panel de navegación y haga clic en **Create Service** en la esquina superior derecha.
- Paso 3** Configure los parámetros relacionados.
- **Service Name:** Especifique un nombre de Service, que puede ser el mismo que el nombre de la carga de trabajo.

- **Access Type:** Seleccione **DNAT**.
- **Namespace:** Espacio de nombres al que pertenece la carga de trabajo.
- **Service Affinity:** Para más información, véase [externalTrafficPolicy \(afinidad del Service\)](#).
  - **Cluster level:** Las direcciones IP y los puertos de acceso de todos los nodos de un clúster se pueden utilizar para acceder a la carga de trabajo asociada con el Service. El acceso al Service causará una pérdida de rendimiento debido a la redirección de la ruta y no se puede obtener la dirección IP de origen del cliente.
  - **Node level:** Solo la dirección IP y el puerto de acceso del nodo donde se encuentra la carga de trabajo pueden acceder a la carga de trabajo asociada con el Service. El acceso al Service no causará pérdida de rendimiento debido a la redirección de la ruta, y se puede obtener la dirección IP de origen del cliente.
- **Selector:** Agregue una etiqueta y haga clic en **Add**. Un Service selecciona un pod basado en la etiqueta agregada. También puede hacer clic en **Reference Workload Label** para hacer referencia a la etiqueta de una carga de trabajo existente. En el cuadro de diálogo que se muestra, seleccione una carga de trabajo y haga clic en **OK**.
- **IPv6:** Esta función está deshabilitada por defecto. Una vez habilitada esta función, la dirección IP del clúster del Service cambia a una dirección IPv6. Para obtener más información, consulte [¿Cómo creo un clúster de doble pila IPv4/IPv6? Este parámetro solo está disponible en clústeres de v1.15 o posterior con IPv6 habilitado \(establecido durante la creación del clúster\)](#).
- **DNAT:** Seleccione el gateway de DNAT y EIP creados en [Creación de un gateway de NAT y una dirección IP elástica](#).
- **Port**
  - **Protocol:** protocolo utilizado por el Service.
  - **Container Port:** puerto en el que escucha la carga de trabajo. La carga de trabajo de Nginx escucha en el puerto 80.
  - **Service Port:** un puerto asignado al puerto contenedor en la dirección IP interna del clúster. Se puede acceder a la carga de trabajo desde <cluster-internal IP address>:<access port>. El rango de número de puerto es de 1–65535.

**Paso 4** Haga clic en **OK**.

----Fin

## Configuración del tipo de acceso con kubectl

Puede establecer el Service al crear una carga de trabajo con kubectl. Esta sección utiliza una carga de trabajo de Nginx como ejemplo para describir cómo implementar el acceso intracluster usando kubectl.

**Paso 1** Utilice kubectl para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).

**Paso 2** Cree y edite los archivos **nginx-deployment.yaml** y **nginx-nat-svc.yaml**.

Los nombres de archivo están definidos por el usuario. **nginx-deployment.yaml** y **nginx-nat-svc.yaml** son simplemente nombres de archivo de ejemplo.

**vi nginx-deployment.yaml**

```
apiVersion: apps/v1
kind: Deployment
```

```

metadata:
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx:latest
        name: nginx
        imagePullSecrets:
        - name: default-secret
    
```

Para ver las descripciones de los campos anteriores, consulte [Tabla 5-2](#).

### vi nginx-nat-svc.yaml

```

apiVersion: v1
kind: Service
metadata:
  name: nginx
  annotations:
    kubernetes.io/elb.class: dnat
    kubernetes.io/natgateway.id: e4a1cfcf-29df-4ab8-a4ea-c05dc860f554
spec:
  loadBalancerIP: 10.78.42.242
  ports:
  - name: service0
    port: 80
    protocol: TCP
    targetPort: 80
  selector:
    app: nginx
  type: LoadBalancer
    
```

**Tabla 7-22** Parámetros de clave

Parámetro	Obligatorio	Tipo	Descripción
kubernetes.io/elb.class	Sí	String	Este parámetro se establece en <b>dnat</b> por lo que CCE puede trabajar con un gateway de NAT y se pueden agregar reglas de DNAT.
kubernetes.io/natgateway.id	Sí	String	ID de un gateway de NAT.
loadBalancerIP	Sí	String	ID de EIP.
port	Sí	Integer	Puerto de acceso establecido en la consola. El valor varía de 1 a 65535.
targetPort	Sí	String	El puerto del contenedor está configurado en la consola. El valor varía de 1 a 65535.
type	Sí	String	El tipo de servicio de gateway de NAT debe establecerse en <b>LoadBalancer</b> .

**Paso 3** Cree una carga de trabajo.

**kubectl create -f nginx-deployment.yaml**

Si se muestra la información similar a la siguiente, se está creando la carga de trabajo.

```
deployment "nginx" created
```

**kubectl get po**

Si se muestra la información similar a la siguiente, la carga de trabajo se está ejecutando.

NAME	READY	STATUS	RESTARTS	AGE
nginx-2601814895-sf71t	1/1	Running	0	8s

**Paso 4** Cree un Service.

**kubectl create -f nginx-nat-svc.yaml**

Si se muestra la información similar a la siguiente, se ha creado el Service.

```
service "nginx-eip" created
```

**kubectl get svc**

Si se muestra la siguiente información, el Service se ha establecido correctamente y se puede acceder a la carga de trabajo.

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.247.0.1	<none>	443/TCP	3d
nginx-nat	LoadBalancer	10.247.226.2	10.154.74.98	80:30589/TCP	5s

**Paso 5** En la barra de direcciones de su navegador, introduzca **10.154.74.98:80** y pulse **Enter**.

En este ejemplo, **10.154.74.98** es la dirección IP elástica y **80** es el número de puerto obtenido en el paso anterior.

----Fin

## 7.3.6 Headless Service

The preceding types of Services allow internal and external pod access, but not the following scenarios:

- Accessing all pods at the same time
- Pods in a Service accessing each other

This is where headless Service come into service. A headless Service does not create a cluster IP address, and the DNS records of all pods are returned during query. In this way, the IP addresses of all pods can be queried. **StatefulSets** use headless Services to support mutual access between pods.

```
apiVersion: v1
kind: Service      # Object type (Service)
metadata:
  name: nginx-headless
  labels:
    app: nginx
spec:
  ports:
    - name: nginx      # - name: nginx      # Name of the port for
      communication between pods
```

```
port: 80          # Port number for communication between pods
selector:
  app: nginx      # Select the pod whose label is app:nginx.
clusterIP: None  # Set this parameter to None, indicating that a headless
Service is to be created.
```

Run the following command to create a headless Service:

```
# kubectl create -f headless.yaml
service/nginx-headless created
```

After the Service is created, you can query the Service.

```
# kubectl get svc
NAME                TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
nginx-headless     ClusterIP    None         <none>        80/TCP     5s
```

Create a pod to query the DNS. You can view the records of all pods. In this way, all pods can be accessed.

```
$ kubectl run -i --tty --image tutum/dnsutils dnsutils --restart=Never --
rm /bin/sh
If you do not see a command prompt, try pressing Enter.
/ # nslookup nginx-0.nginx
Server:          10.247.3.10
Address:         10.247.3.10#53
Name:   nginx-0.nginx.default.svc.cluster.local
Address: 172.16.0.31

/ # nslookup nginx-1.nginx
Server:          10.247.3.10
Address:         10.247.3.10#53
Name:   nginx-1.nginx.default.svc.cluster.local
Address: 172.16.0.18

/ # nslookup nginx-2.nginx
Server:          10.247.3.10
Address:         10.247.3.10#53
Name:   nginx-2.nginx.default.svc.cluster.local
Address: 172.16.0.19
```

## 7.4 Ingresos

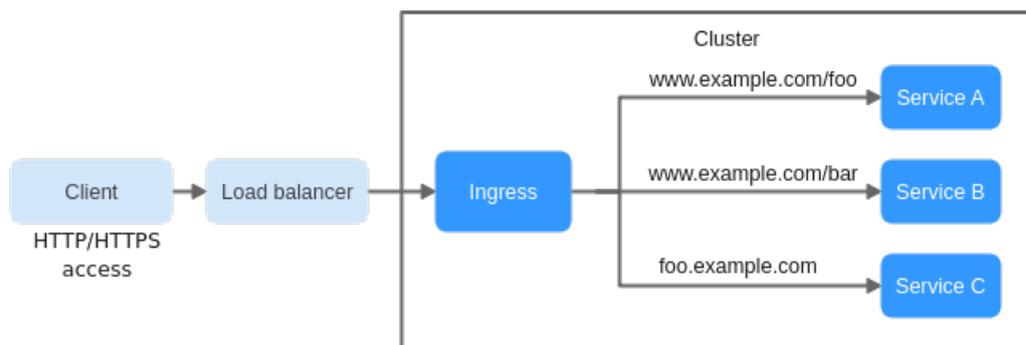
### 7.4.1 Descripción de entrada

#### Por qué necesitamos entradas

Un Service se utiliza generalmente para reenviar solicitudes de acceso basadas en TCP y UDP y proporcionar balanceo de carga de capa 4 para clústeres. Sin embargo, en escenarios reales, si hay un gran número de solicitudes de acceso HTTP/HTTPS en la capa de aplicación, el Service no puede cumplir con los requisitos de reenvío. Por lo tanto, el clúster de Kubernetes proporciona un modo de acceso basado en HTTP, es decir, entrada.

Una entrada es un recurso independiente en el clúster de Kubernetes y define reglas para reenviar el tráfico de acceso externo. Como se muestra en [Figura 7-27](#), puede personalizar las reglas de reenvío basadas en nombres de dominio y URL para implementar una distribución detallada del tráfico de acceso.

**Figura 7-27** Diagrama de entrada



A continuación se describen las definiciones relacionadas con la entrada:

- **Objeto de entrada:** un conjunto de reglas de acceso que reenvía solicitudes a los servicios especificados en función de nombres de dominio o direcciones URL. Puede agregarse, eliminarse, modificarse y consultarse invocando a las API.
- **Ingress Controller:** un ejecutor para el reenvío de solicitudes. Supervisa los cambios de objetos de recursos como entradas, Services, puntos de conexión, secretos (principalmente certificados y claves TLS), nodos y ConfigMaps en tiempo real, analiza las reglas definidas por entradas, y reenvía solicitudes a los Services de backend correspondientes.

Los Ingress Controllers proporcionados por diferentes proveedores se implementan de diferentes maneras. Basado en los tipos de balanceadores de carga, Ingress Controllers se clasifican en ELB Ingress Controller y Nginx Ingress Controller. Ambos son compatibles con CCE. ELB Ingress Controller reenvía el tráfico con ELB. Nginx Ingress Controller utiliza las plantillas e imágenes mantenidas por la comunidad de Kubernetes para reenviar el tráfico a través del componente de Nginx.

## Comparación de características de entrada

**Tabla 7-23** Comparación entre las características de entrada

Función	ELB Ingress Controller	Nginx Ingress Controller
O&M	Sin O&M	Autoinstalación, actualización y mantenimiento
Rendimiento	Una entrada solo soporta un balanceador de carga.	Múltiples entradas soportan un balanceador de carga.
	Los balanceadores de carga de nivel empresarial se utilizan para proporcionar un alto rendimiento y una alta disponibilidad. El reenvío de Service no se ve afectado en los escenarios de actualización y fallo.	El rendimiento varía en función de la configuración de recursos de los pods.

Función	ELB Ingress Controller	Nginx Ingress Controller
	Se admite la carga dinámica.	Se requiere volver a cargar después de actualizar las configuraciones, lo que puede interrumpir los servicios.
Despliegue de componentes	Desplegado en el nodo principal	Desplegado en los nodos de trabajo y los costos de operación requeridos para el componente de Nginx
Redirección de rutas	No se admite	Se admite
Configuración de SSL	Se admite	Se admite
Uso de entrada como proxy para servicios de backend	No se admite	Soportado, que se puede implementar con backend-protocolo: anotaciones de HTTPS.

ELB Ingress es esencialmente diferente del Nginx Ingress de código abierto. Por lo tanto, sus tipos de Service soportados también son diferentes.

ELB Ingress Controller se despliega en un nodo principal. Todas las políticas y comportamientos de reenvío se configuran en el lado de ELB. Los balanceadores de carga fuera del clúster solo pueden conectarse a los nodos del clúster con la dirección IP de VPC en escenarios de redes sin paso. Por lo tanto, ELB Ingress solo admite los Services de NodePort. Sin embargo, en el escenario de redes de paso (cluster de Turbo de CCE + balanceador de carga dedicado), ELB puede reenviar directamente el tráfico a los pods del clúster. En este caso, además de los Services de NodePort, la entrada también puede interconectarse con los Services de ClusterIP.

Nginx Ingress Controller se ejecuta en un clúster y se expone como un Service con NodePort. El tráfico se reenvía a otros Services en el clúster con Nginx-ingress. El comportamiento de reenvío de tráfico y el objeto de reenvío están en el clúster. Por lo tanto, se admiten los Services de ClusterIP y de NodePort.

En conclusión, ELB Ingress utiliza balanceadores de carga de nivel empresarial para reenviar el tráfico y ofrece un alto rendimiento y estabilidad. Nginx Ingress Controller se despliega en nodos de clúster, que consume recursos de clúster, pero tiene mejor capacidad de configuración.

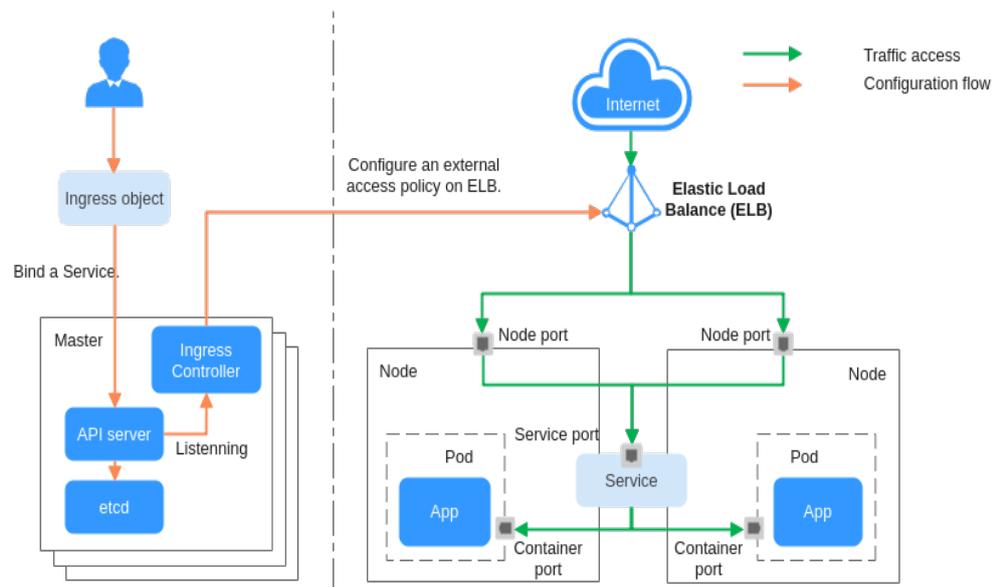
## Principio de funcionamiento del ELB Ingress Controller

ELB Ingress Controller desarrollado por CCE implementa el acceso a la red de capa 7 para Internet e Intranet (en la misma VPC) basado en ELB y distribuye el tráfico de acceso a los Services correspondientes usando diferentes URLs.

ELB Ingress Controller se despliega en el nodo principal y se vincula al balanceador de carga en la VPC donde reside el clúster. Se pueden configurar diferentes nombres de dominio, puertos y políticas de reenvío para el mismo balanceador de carga (con la misma dirección IP). [Figura 7-28](#) muestra el principio de funcionamiento del ELB Ingress Controller.

1. Un usuario crea un objeto de ingreso y configura una regla de acceso de tráfico en el ingreso, incluidos el balanceador de carga, el URL, SSL y el puerto de servicio de backend.
2. Cuando Ingress Controller detecta que el objeto de entrada cambia, reconfigura el oyente y la ruta de servidor backend en el lado de ELB según la regla de acceso de tráfico.
3. Cuando un usuario accede a una carga de trabajo, el tráfico se reenvía al puerto de Service backend correspondiente basándose en la política de reenvío configurada en ELB y, a continuación, se reenvía a cada carga de trabajo asociada a través del Service.

**Figura 7-28** Principio de funcionamiento del ELB Ingress Controller



## Principio de funcionamiento del Nginx Ingress Controller

Una entrada de Nginx utiliza ELB como entrada de tráfico. El complemento **nginx-ingress** se despliega en un clúster para equilibrar el tráfico y controlar el acceso.

### 📖 NOTA

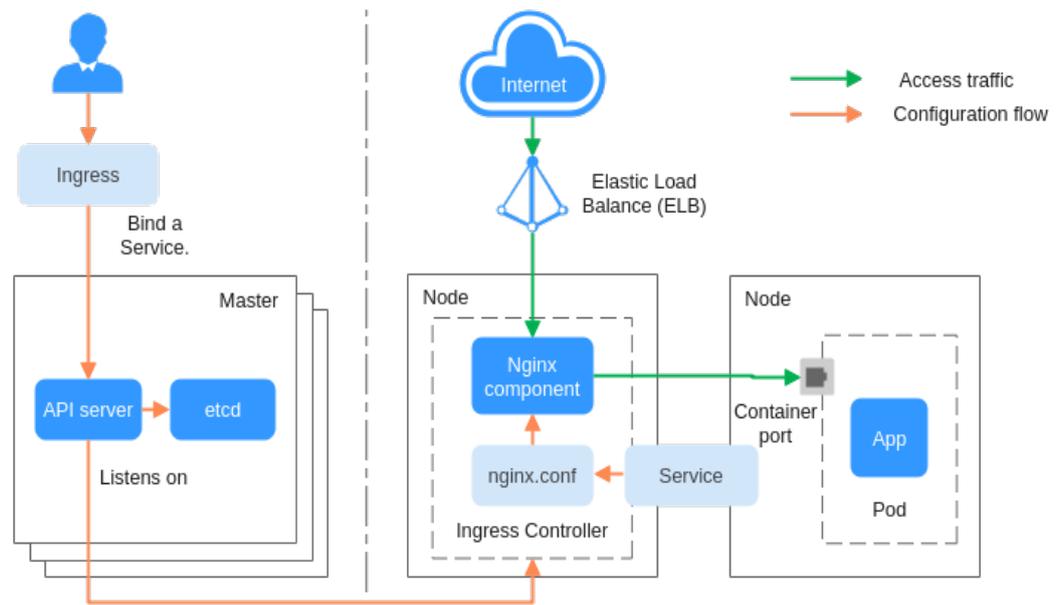
El complemento de **nginx-ingress** en CCE se implementa usando el gráfico y la imagen de comunidad de código abierto. CCE no mantiene el complemento. Por lo tanto, no se recomienda que el complemento de **nginx-ingress** se use comercialmente.

Puede visitar la [comunidad de código abierto](#) para obtener más información.

Nginx Ingress Controller se despliega en los nodos de trabajo con pods, lo que resultará en costos de operación y gastos generales de ejecución del componente de Nginx. **Figura 7-29** muestra los principios de funcionamiento de Nginx Ingress Controller.

1. Después de actualizar los recursos de ingreso, Nginx Ingress Controller escribe una regla de reenvío definida en los recursos de ingreso en el archivo de configuración **nginx.conf** de Nginx.
2. El componente de Nginx integrado recarga el archivo de configuración actualizado para modificar y actualizar la regla de reenvío de Nginx.
3. Cuando el tráfico accede a un clúster, el balanceador de carga creado primero reenvía el tráfico al componente de Nginx en el clúster. A continuación, el componente de Nginx reenvía el tráfico a cada carga de trabajo basándose en la regla de reenvío.

**Figura 7-29** Principio de funcionamiento del Nginx Ingress Controller



## 7.4.2 Ingreso de ELB

### 7.4.2.1 Creación de ELB Ingress en la consola

#### Requisitos previos

- Un ingreso proporciona acceso a la red para cargas de trabajo backend. Asegúrese de que una carga de trabajo esté disponible en un clúster. Si no hay ninguna carga de trabajo disponible, despliegue una carga de trabajo haciendo referencia a [Creación de un Deployment](#), [Creación de un StatefulSet](#) o [Creación de un DaemonSet](#).
- Los balanceadores de carga dedicados deben ser del tipo de aplicación (HTTP/HTTPS) que admita las redes privadas (con una IP privada).
- En las redes de paso a través de ELB (clúster de Turbo de CCE + balanceador de carga dedicado), ELB Ingress admite servicios ClusterIP. En otros escenarios, ELB Ingress admite los servicios de NodePort.

#### Precauciones

- Se recomienda que otros recursos no utilicen el balanceador de carga creado automáticamente por un ingreso. De lo contrario, el balanceador de carga estará ocupado cuando se elimine la entrada, dando como resultado recursos residuales.
- Después de crear una entrada, actualice y mantenga la configuración de los balanceadores de carga seleccionados en la consola de CCE. No modifique la configuración en la consola de ELB. De lo contrario, el servicio de entrada puede ser anormal.
- El URL registrado en una política de reenvío de ingreso debe ser la misma que la dirección URL utilizada para acceder al Service de backend. De lo contrario, se devolverá un error 404.
- En un clúster que usa el modo proxy IPVS, si el ingreso y el Service usan el mismo balanceador de carga de ELB, no se puede acceder a la entrada desde los nodos y

contenedores en el clúster porque kube-proxy monta la dirección de LoadBalancer Service en el puente ipvs-0. Este puente intercepta el tráfico del balanceador de carga conectado a la entrada. Se recomienda utilizar diferentes balanceadores de carga de ELB para la entrada y el Service.

- No conecte el ingreso y el **Service usando HTTP** al mismo oyente del mismo balanceador de carga. De lo contrario, se produce un conflicto de puerto.

## Adición de un ingreso de ELB

Esta sección utiliza una carga de trabajo de Nginx como ejemplo para describir cómo agregar un ingreso de ELB.

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** Elija **Networking** en el panel de navegación, haga clic en la ficha **Ingresses** y haga clic en **Create Service** en la esquina superior derecha.

**Paso 3** Establezca los parámetros de entrada.

- **Name:** Especifique el nombre de una entrada, por ejemplo, **ingress-demo**.
- **Interconnect with Nginx:** Esta opción solo se muestra después de instalar el complemento **nginx-ingress**. Si esta opción está disponible, se ha instalado el complemento nginx-ingress. Al activar esta opción se creará una entrada de Nginx. Desactívela si desea crear una entrada de ELB. Para obtener más información, véase **Creación de entradas de Nginx en la consola**.

- **Load Balancer**

Seleccione el balanceador de carga que desea interconectar. Solo se admiten balanceadores de carga en la misma VPC que el clúster. Si no hay ningún balanceador de carga disponible, haga clic en **Create Load Balancer** para crear uno en la consola de ELB.

Los balanceadores de carga dedicados deben admitir HTTP y el tipo de red debe admitir las redes privadas.

La consola de CCE admite la creación automática de balanceadores de carga. Seleccione **Auto create** en el cuadro de lista desplegable y establezca los siguientes parámetros:

- **Instance Name:** Ingrese un nombre de balanceador de carga.
- **Public Access:** Si está habilitado, se creará una EIP con ancho de banda de 5 Mbit/s. De forma predeterminada, se cobra por el tráfico.
- **AZ, Subnet y Specifications** (disponible solo para balanceadores de carga dedicados): Establezca la AZ, la subred y las especificaciones. Actualmente, solo se pueden crear automáticamente los balanceadores de carga dedicados del tipo de red (TCP/UDP).
- **Listener Configuration:** Ingreso configura un oyente para el balanceador de carga, que escucha las solicitudes del balanceador de carga y distribuye el tráfico. Una vez completada la configuración, se crea un oyente en el balanceador de carga. El nombre de oyente predeterminado es *k8s\_<Protocol type>\_<Port number>*, por ejemplo, *k8s\_HTTP\_80*.
  - **Front-End Protocol:** HTTP y HTTPS están disponibles.
  - **External Port:** Número de puerto abierto a la dirección de servicio de ELB. El número de puerto se puede especificar aleatoriamente.
  - **Certificate Source:** Se admite el secreto de TLS y el certificado de servidor de ELB.

- **Server Certificate:** Cuando se crea un oyente de HTTPS para un balanceador de carga, debe vincular un certificado al balanceador de carga para admitir la autenticación cifrada para la transmisión de datos de HTTPS.
  - **TLS secret:** Para obtener más información sobre cómo crear un certificado secreto, consulte [Creación de un secreto](#).
  - **ELB server certificate:** Utilice el certificado creado en el servicio de ELB.

 **NOTA**

Si ya hay un ingreso de HTTPS para el puerto elegido en el balanceador de carga, el certificado del nuevo ingreso de HTTPS debe ser el mismo que el certificado del ingreso existente. Esto significa que un oyente solo tiene un certificado. Si se agregan dos certificados, cada uno con una entrada diferente, al mismo oyente del mismo balanceador de carga, solo el certificado agregado más temprano tiene efecto en el balanceador de carga.

- **SNI:** Server Name Indication (SNI) es un protocolo extendido de TLS. Permite proporcionar múltiples nombres de dominio de acceso basados en TLS para sistemas externos que utilizan la misma dirección IP y puerto. Diferentes nombres de dominio pueden utilizar diferentes certificados de seguridad. Después de habilitar el SNI, el cliente puede enviar el nombre de dominio solicitado al iniciar una solicitud de handshake de TLS. Después de recibir la solicitud de TLS, el balanceador de carga busca el certificado basado en el nombre de dominio en la solicitud. Si se encuentra el certificado correspondiente al nombre de dominio, el balanceador de carga devuelve el certificado para la autorización. De lo contrario, se devuelve el certificado predeterminado (certificado de servidor) para la autorización.

 **NOTA**

- La opción **SNI** solo está disponible cuando se selecciona **HTTPS**.
  - Esta función solo se admite para clústeres de v1.15.11 y posteriores.
  - Especifique el nombre de dominio para el certificado de SNI. Solo se puede especificar un nombre de dominio para cada certificado. Se admiten los certificados de dominio comodín.
- **Forwarding Policies:** Cuando la dirección de acceso de una solicitud coincide con la política de reenvío (una política de reenvío consiste en un nombre de dominio y un URL, por ejemplo, 10.117.117.117:80/helloworld), la solicitud se reenvía al Service de destino correspondiente para su procesamiento. Puede hacer clic en  para agregar varias políticas de reenvío.
  - **Domain Name:** nombre de dominio real. Asegúrese de que el nombre de dominio ha sido registrado y archivado. Una vez configurada una regla de nombre de dominio, debe usar el nombre de dominio para tener acceso.
  - Regla de coincidencia para direcciones URL
    - **Prefix match:** Si el URL está establecido en `/healthz`, se puede acceder al URL que cumple con el prefijo. Por ejemplo, `/healthz/v1` y `/healthz/v2`.
    - **Exact match:** Se puede acceder al URL solo cuando está totalmente coincidente. Por ejemplo, si el URL está establecido en `/healthz`, solo se puede acceder a `/healthz`.
    - **Regular expression:** El URL se hace coincidir en función de la expresión regular. Por ejemplo, si la expresión regular es `/[A-Za-z0-9_-]+/test`, se puede acceder a todos los URL que cumplan con esta regla, por ejemplo, `/abcA9/test` y `/v1-Ab/test`. Se admiten dos estándares de expresión regular: POSIX y Perl.

- **URL:** ruta de acceso a registrar, por ejemplo, **/healthz**.

 **NOTA**

La ruta de acceso agregada aquí debe existir en la aplicación de backend. De lo contrario, el reenvío falla.

Por ejemplo, el URL de acceso predeterminado de la aplicación Nginx es **/usr/share/nginx/html**. Al agregar **/test** a la política de reenvío de ingreso, asegúrese de que su aplicación de Nginx contiene el mismo URL, es decir, **/usr/share/nginx/html/test**, de lo contrario, se devuelve 404.

- **Destination Service:** Seleccione un Service existente o cree un Service. Los Service que no cumplen los criterios de búsqueda se eliminan automáticamente.
- **Destination Service Port:** Seleccione el puerto de acceso del Service de destino.
- **Set ELB:**
  - **Distribution Policy:** Hay tres algoritmos disponibles: Round robin ponderado, algoritmo de conexiones mínimas ponderadas o hash de IP de origen.

 **NOTA**

- **Round robin ponderado:** las solicitudes se reenvían a diferentes servidores en función de sus pesos, lo que indica el rendimiento del procesamiento del servidor. Los servidores backend con mayores pesos reciben proporcionalmente más solicitudes, mientras que los servidores con igual peso reciben el mismo número de solicitudes. Este algoritmo se utiliza a menudo para las conexiones cortas, como los servicios de HTTP.
- **Conexiones mínimas ponderadas:** Además del peso asignado a cada servidor, también se considera el número de conexiones procesadas por cada servidor backend. Las solicitudes se reenvían al servidor con la relación de conexiones/peso más baja. Basado en las **conexiones mínimas**, el algoritmo **conexiones mínimas ponderadas** asigna un peso a cada servidor basado en su capacidad de procesamiento. Este algoritmo se utiliza a menudo para las conexiones persistentes, tales como las conexiones de base de datos.
- **Hash de IP de origen:** La dirección IP de origen de cada solicitud se calcula usando el algoritmo hash para obtener una clave hash única, y todos los servidores backend están numerados. La clave generada asigna el cliente a un servidor determinado. Esto permite que las solicitudes de diferentes clientes se distribuyan en modo de equilibrio de carga y garantiza que las solicitudes del mismo cliente se reenvíen al mismo servidor. Este algoritmo se aplica a las conexiones de TCP sin cookies.
- **Sticky Session:** Esta función está deshabilitada por defecto. Las opciones son las siguientes:
  - **Load balancer cookie:** Ingrese la **Stickness Duration** que va de 1 a 1,440 minutos.
  - **Application cookie:** Este parámetro solo está disponible para los balanceadores de carga compartidos. Además, debe introducir **Cookie Name** que oscila entre 1 y 64 caracteres.

 **NOTA**

Cuando la **política de distribución** utiliza el hash IP de origen, no se puede establecer la sesión adhesiva.

- **Health Check:** Establezca la configuración de comprobación de estado del balanceador de carga. Si esta función está habilitada, se admiten las siguientes configuraciones:

Parámetro	Descripción
Protocol	<p>Cuando el protocolo del puerto de servicio de destino se establece en TCP, se admiten TCP y HTTP. Cuando se establece en UDP, solo se admite UDP.</p> <ul style="list-style-type: none"> <li>○ <b>Check Path</b> (soportado únicamente por el protocolo de comprobación de estado HTTP): especifica el URL de comprobación de estado. La ruta de comprobación debe comenzar con un (/) de barra diagonal y contener entre 1 y 80 caracteres.</li> </ul>
Port	<p>De forma predeterminada, el puerto de Service (Puerto de nodo y puerto contenedor del Service) se utiliza para la comprobación de estado. También puede especificar otro puerto para la comprobación de estado. Después de especificar el puerto, se agregará un puerto de servicio llamado <b>cce-healthz</b> para el Service.</p> <ul style="list-style-type: none"> <li>○ <b>Node Port</b>: si se utiliza un balanceador de carga compartido o no se asocia ninguna instancia ENI, el puerto de nodo se utiliza como puerto de comprobación de estado. Si no se especifica este parámetro, se utiliza un puerto aleatorio. El valor oscila entre 30000 y 32767.</li> <li>○ <b>Container Port</b>: Cuando un balanceador de carga dedicado está asociado a una instancia ENI, el puerto contenedor se utiliza para la comprobación de estado. El valor varía de 1 a 65535.</li> </ul>
Check Interval (s)	Especifica el intervalo máximo entre las comprobaciones de estado. El valor varía de 1 a 50.
Timeout (s)	Especifica la duración máxima del tiempo de espera para cada comprobación de estado. El valor varía de 1 a 50.
Max. Retries	Especifica el número máximo de reintentos de comprobación de estado. El valor varía de 1 a 10.

– **Operation**: Haga clic en **Delete** para eliminar la configuración.

- **Annotation**: Las entradas proporcionan algunas funciones avanzadas de CCE, que se implementan mediante anotaciones. Cuando se usa kubectl para crear un contenedor se usarán anotaciones. Para más detalles, véase [Creación de una entrada - Creación automática de un balanceador de carga](#) y [Creación de un ingreso - interconexión con un balanceador de carga existente](#).

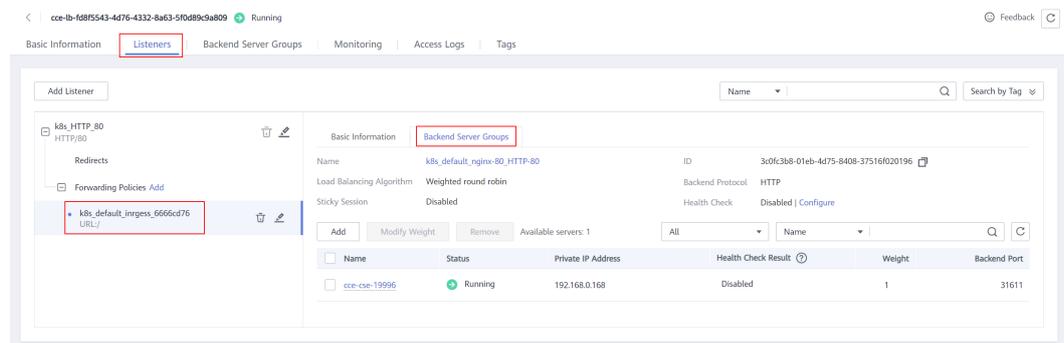
**Paso 4** Una vez completada la configuración, haga clic en **OK**. Después de crear la entrada, se muestra en la lista de entrada.

En la consola de ELB, puede ver el ELB creado automáticamente con CCE. El nombre predeterminado es **cce-lb-ingress.UID**. Haga clic en el nombre del ELB para acceder a su página de detalles. En la página de ficha **Listeners**, vea la configuración de la ruta del ingreso, incluidos el URL, el puerto de oyente y el puerto del grupo de servidores backend.

**AVISO**

Después de crear la entrada, actualice y mantenga el balanceador de carga seleccionado en la consola de CCE. No mantenga el balanceador de carga en la consola de ELB. De lo contrario, el servicio de entrada puede ser anormal.

**Figura 7-30** Configuración de enrutamiento de ELB



**Paso 5** Acceda a la interfaz /healthz de la carga de trabajo, por ejemplo, carga de trabajo defaultbackend.

1. Obtenga la dirección de acceso de la interfaz /healthz de la carga de trabajo. La dirección de acceso consiste en la dirección IP del balanceador de carga, el puerto externo y el URL de asignación, por ejemplo, 10.\*\*.\*\*.\*/80/healthz.
2. Ingrese el URL de la interfaz /healthz, por ejemplo http://10\*\*.\*\*.\*/80/healthz, en el cuadro de direcciones del navegador para acceder a la carga de trabajo, como se muestra en [Figura 7-31](#).

**Figura 7-31** Acceso a la interfaz /healthz de defaultbackend



----Fin

## Operaciones relacionadas

La estructura de entrada de Kubernetes no contiene el campo **property**. Por lo tanto, la entrada creada por la API invocada por client-go no contiene el campo **property**. CCE proporciona una solución para garantizar la compatibilidad con Kubernetes client-go. Para obtener más información sobre la solución, consulte [¿Cómo puedo lograr la compatibilidad entre la propiedad de Ingress y el cliente-go de Kubernetes?](#)

## 7.4.2.2 Uso de kubectl para crear una entrada de ELB

### Escenario

Esta sección utiliza una [carga de trabajo de Nginx](#) como ejemplo para describir cómo crear una entrada ELB usando kubectl.

- Si no hay ningún balanceador de carga disponible en la misma VPC, CCE puede crear automáticamente un balanceador de carga al crear una entrada. Para obtener más información, véase [Creación de una entrada - Creación automática de un balanceador de carga](#).
- Si un balanceador de carga está disponible en la misma VPC, realice la operación haciendo referencia a [Creación de un ingreso - interconexión con un balanceador de carga existente](#).

### Requisitos previos

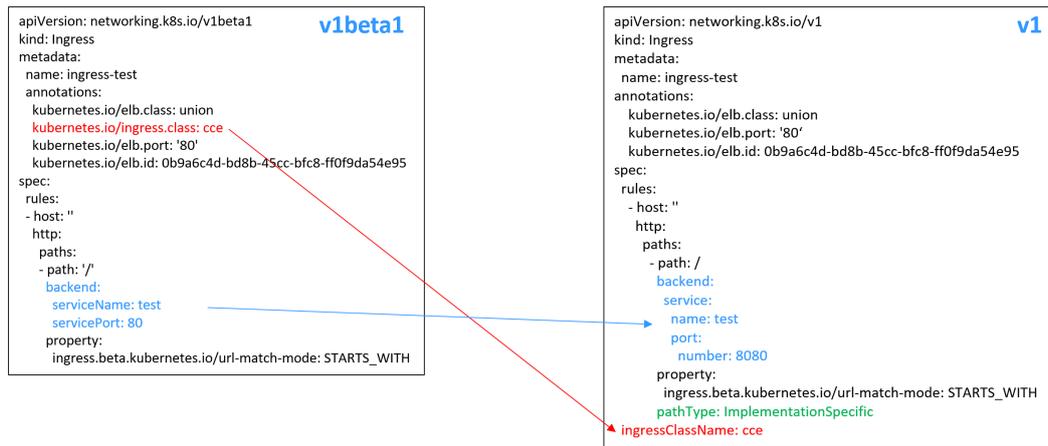
- Un ingreso proporciona acceso a la red para cargas de trabajo back-end. Asegúrese de que una carga de trabajo esté disponible en un clúster. Si no hay ninguna carga de trabajo disponible, despliegue un ejemplo de la carga de trabajo de Nginx haciendo referencia a [Creación de una Deployment](#), [Creación de un StatefulSet](#) o [Creación de un DaemonSet](#).
- Se ha configurado un Service de NodePort para la carga de trabajo. Para obtener más información acerca de cómo configurar el Service, consulte [NodePort](#).
- Los balanceadores de carga dedicados deben ser del tipo de aplicación (HTTP/HTTPS) que admita las redes privadas (con una IP privada).

### Ingress Description of networking.k8s.io/v1

En los clústeres de CCE de v1.23 o posterior, la versión de ingreso se cambia a [networking.k8s.io/v1](#).

Comparado con v1beta1, v1 tiene las siguientes diferencias en parámetros:

- El tipo de entrada se cambia de [kubernetes.io/ingress.class](#) en [annotations](#) a [spec.ingressClassName](#).
- Se cambia el formato de [backend](#).
- El parámetro [pathType](#) debe especificarse para cada ruta. Las opciones son las siguientes:
  - **ImplementationSpecific**: El método de coincidencia depende del controlador de entrada. El método de coincidencia definido por [ingress.beta.kubernetes.io/url-match-mode](#) se usa en CCE, que es el mismo que v1beta1.
  - **Exact**: coincidencia exacta del URL, que distingue entre mayúsculas y minúsculas.
  - **Prefix**: coincidencia basada en el prefijo de URL separado por una barra diagonal (/). La coincidencia distingue entre mayúsculas y minúsculas, y los elementos de la ruta se hacen coincidir uno por uno. Un elemento de trazado hace referencia a una lista de etiquetas en el trazado separadas por una barra diagonal (/).



## Creación de una entrada - Creación automática de un balanceador de carga

A continuación se describe cómo ejecutar el comando `kubectl` para crear automáticamente un balanceador de carga al crear una entrada.

**Paso 1** Utilice `kubectl` para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).

**Paso 2** Cree un archivo YAML denominado `ingress-test.yaml`. El nombre del archivo se puede personalizar.

vi `ingress-test.yaml`

### NOTA

A partir del clúster v1.23, la versión de ingreso cambia de `networking.k8s.io/v1beta1` a `networking.k8s.io/v1`. Para obtener más información sobre las diferencias entre v1 y v1beta1, consulte [Ingress Description of networking.k8s.io/v1](#).

### Ejemplo de un balanceador de carga compartido (acceso a red pública) para clústeres de v1.23 o posterior:

```

apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress-test
  annotations:
    kubernetes.io/elb.class: union
    kubernetes.io/elb.port: '80'
    kubernetes.io/elb.autocreate:
      '{
        "type": "public",
        "bandwidth_name": "cce-bandwidth-*****",
        "bandwidth_chargemode": "bandwidth",
        "bandwidth_size": 5,
        "bandwidth_sharetype": "PER",
        "eip_type": "5_bgp"
      }'
spec:
  rules:
  - host: ''
    http:
      paths:
      - path: '/'
        backend:
          service:
            name: <your_service_name> # Replace it with the name of your target
            Service.
    
```

```

        port:
          number: <your_service_port> # Replace it with the port number of
your target Service.
        property:
          ingress.beta.kubernetes.io/url-match-mode: STARTS_WITH
          pathType: ImplementationSpecific
ingressClassName: cce # ELB ingress is used.
    
```

### Ejemplo de un balanceador de carga compartido (acceso a red pública) para clústeres de v1.21 o anteriores:

```

apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: ingress-test
  annotations:
    kubernetes.io/elb.class: union
    kubernetes.io/ingress.class: cce # ELB ingress is used.
    kubernetes.io/elb.port: '80'
    kubernetes.io/elb.autocreate:
      '{
        "type": "public",
        "bandwidth_name": "cce-bandwidth-*****",
        "bandwidth_chargemode": "bandwidth",
        "bandwidth_size": 5,
        "bandwidth_sharetype": "PER",
        "eip_type": "5_bgp"
      }'
spec:
  rules:
  - host: ''
    http:
      paths:
      - path: '/'
        backend:
          serviceName: <your_service_name> # Replace it with the name of your
target Service.
          servicePort: <your_service_port> # Replace it with the port number of
your target Service.
        property:
          ingress.beta.kubernetes.io/url-match-mode: STARTS_WITH
    
```

### Ejemplo de un balanceador de carga dedicado (acceso a red pública) para clústeres de v1.23 o posterior:

```

apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress-test
  namespace: default
  annotations:
    kubernetes.io/elb.class: performance
    kubernetes.io/elb.port: '80'
    kubernetes.io/elb.autocreate:
      '{
        "type": "public",
        "bandwidth_name": "cce-bandwidth-*****",
        "bandwidth_chargemode": "bandwidth",
        "bandwidth_size": 5,
        "bandwidth_sharetype": "PER",
        "eip_type": "5_bgp",
        "available_zone": [
          "ap-southeast-1a"
        ],
        "l7_flavor_name": "L7_flavor.elb.s1.small"
      }'
spec:
  rules:
  - host: ''
    http:
      paths:
    
```

```

- path: '/'
  backend:
    service:
      name: <your_service_name> # Replace it with the name of your target
Service.
      port:
        number: <your_service_port> # Replace it with the port number of
your target Service.
      property:
        ingress.beta.kubernetes.io/url-match-mode: STARTS_WITH
        pathType: ImplementationSpecific
    ingressClassName: cce
    
```

**Ejemplo de un balanceador de carga dedicado (acceso a la red pública) para clústeres de la versión 1.21 o anterior:**

```

apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: ingress-test
  namespace: default
  annotations:
    kubernetes.io/elb.class: performance
    kubernetes.io/ingress.class: cce
    kubernetes.io/elb.port: '80'
    kubernetes.io/elb.autocreate:
      '{
        "type": "public",
        "bandwidth_name": "cce-bandwidth-*****",
        "bandwidth_chargemode": "bandwidth",
        "bandwidth_size": 5,
        "bandwidth_sharetype": "PER",
        "eip_type": "5_bgp",
        "available_zone": [
          "ap-southeast-1a"
        ],
        "l7_flavor_name": "L7_flavor.elb.s1.small"
      }'
spec:
  rules:
    - host: ''
      http:
        paths:
          - path: '/'
            backend:
              serviceName: <your_service_name> # Replace it with the name of your
target Service.
              servicePort: <your_service_port> # Replace it with the port number of
your target Service.
            property:
              ingress.beta.kubernetes.io/url-match-mode: STARTS_WITH
    
```

**Tabla 7-24** Parámetros de clave

Parámetro	Obligato rio	Tipo	Descripción
kubernetes.io/ elb.class	Sí	String	<p>Seleccione un tipo de balanceador de carga adecuado.</p> <p>El valor puede ser:</p> <ul style="list-style-type: none"> <li>● <b>union</b>: balanceador de carga compartido</li> <li>● <b>performance</b>: balanceador de carga dedicado. Para obtener más información, consulte <a href="#">Diferencias entre los balanceadores de carga compartido y dedicado</a>.</li> </ul> <p>Predeterminado: <b>union</b></p>
kubernetes.io/ ingress.class	Sí (solo para clústeres de v1.21 o anteriores)	String	<p><b>cce</b>: Se utiliza la entrada de ELB autodesarrollada.</p> <p>Este parámetro es obligatorio cuando se crea una entrada invocando a la API.</p>
ingressClassName	Sí (solo para los clústeres de v1.23 o posterior)	String	<p><b>cce</b>: Se utiliza la entrada de ELB autodesarrollada.</p> <p>Este parámetro es obligatorio cuando se crea una entrada invocando a la API.</p>
kubernetes.io/ elb.port	Sí	Integer	<p>Este parámetro indica el puerto externo registrado con la dirección del LoadBalancer Service.</p> <p>Rango soportado: 1 a 65535</p>
kubernetes.io/ elb.subnet-id	-	String	<p>ID de la subred donde se encuentra el clúster. El valor puede contener de 1 a 100 caracteres.</p> <ul style="list-style-type: none"> <li>● Obligatorio cuando se va a crear automáticamente un clúster de v1.11.7-r0 o anterior.</li> <li>● Opcional para los clústeres posteriores a v1.11.7-r0. Se deja en blanco por defecto.</li> </ul> <p>Para obtener más detalles sobre cómo obtener el valor, consulte <a href="#">¿Cuál es la diferencia entre la API de subred de VPC y la API de subred de Neutrón OpenStack?</a></p>

Parámetro	Obligatorio	Tipo	Descripción
kubernetes.io/ elb.enterpriseID	Sí	String	<p><b>Los clústeres de Kubernetes de v1.15 y versiones posteriores admiten este campo. En los clústeres de Kubernetes anteriores a v1.15, los balanceadores de carga se crean en el proyecto predeterminado de forma predeterminada.</b></p> <p>ID del proyecto de empresa en el que se creará el balanceador de carga.</p> <p>El valor contiene de 1 a 100 caracteres.</p> <p><b>Cómo obtenerlo:</b></p> <p>Inicie sesión en la consola de gestión y seleccione <b>Enterprise &gt; Project Management</b> en la barra de menú superior. En la lista que se muestra, haga clic en el nombre del proyecto de empresa de destino y copie el ID en la página de detalles del proyecto de empresa.</p>
kubernetes.io/ elb.autocreate	Sí	elb.autocreate e object	<p>Si se crea automáticamente un balanceador de carga asociado a una entrada. Para obtener más información sobre la descripción del campo, consulte <a href="#">Tabla 7-25</a>.</p> <p><b>Ejemplo</b></p> <ul style="list-style-type: none"> <li>● Si se creará automáticamente un balanceador de carga de red pública, establezca este parámetro en el siguiente valor:  <pre>{ "type": "public", "bandwidth_name": "cce-bandwidth-*****", "bandwidth_chargemode": "bandwidth", "bandwidth_size": 5, "bandwidth_sharetype": "PER", "eip_type": "5_bgp", "name": "james" }</pre> </li> <li>● Si se creará automáticamente un balanceador de carga de red privada, establezca este parámetro en el siguiente valor:  <pre>{ "type": "inner", "name": "A-location-d-test" }</pre> </li> </ul>

Parámetro	Obligatorio	Tipo	Descripción
host	No	String	Nombre de dominio para acceder al Service. De forma predeterminada, este parámetro se deja en blanco y el nombre de dominio debe coincidir completamente. Asegúrese de que el nombre de dominio ha sido registrado y archivado. Una vez configurada una regla de nombre de dominio, debe usar el nombre de dominio para tener acceso.
path	Sí	String	Ruta de ruta definida por el usuario. Todas las solicitudes de acceso externo deben coincidir con <b>host</b> y <b>path</b> . <b>NOTA</b> La ruta de acceso agregada aquí debe existir en la aplicación de backend. De lo contrario, el reenvío falla. Por ejemplo, el URL de acceso predeterminado de la aplicación Nginx es <code>/usr/share/nginx/html</code> . Al agregar <code>/test</code> a la política de reenvío de ingreso, asegúrese de que su aplicación de Nginx contiene el mismo URL, es decir, <code>/usr/share/nginx/html/test</code> , de lo contrario, se devuelve 404.
ingress.beta.kubernetes.io/url-match-mode	No	String	Política de coincidencia de rutas. Predeterminado: <b>STARTS_WITH</b> (Coincidencia de prefijo) Opciones: <ul style="list-style-type: none"> <li>● <b>EQUAL_TO</b>: coincidencia exacta</li> <li>● <b>STARTS_WITH</b>: coincidencia de prefijos</li> <li>● <b>REGEX</b>: coincidencia de expresiones regulares</li> </ul>

Parámetro	Obligatorio	Tipo	Descripción
pathType	Sí	String	<p>Tipo de ruta. Este campo solo es compatible con los clústeres de v1.23 o posterior.</p> <ul style="list-style-type: none"> <li>● <b>ImplementationSpecific</b>: El método de coincidencia depende del controlador de entrada. El método de emparejamiento definido por <b>ingress.beta.kubernetes.io/url-match-mode</b> se usa en CCE.</li> <li>● <b>Exact</b>: coincidencia exacta del URL, que distingue entre mayúsculas y minúsculas.</li> <li>● <b>Prefix</b>: coincidencia basada en el prefijo de URL separado por una barra diagonal (/). La coincidencia distingue entre mayúsculas y minúsculas, y los elementos de la ruta se hacen coincidir uno por uno. Un elemento de trazado hace referencia a una lista de etiquetas en el trazado separadas por una barra diagonal (/).</li> </ul>

**Tabla 7-25** Estructura de datos del campo elb.autocreate

Parámetro	Obligatorio	Tipo	Descripción
type	No	String	<p>Tipo de red del balanceador de carga.</p> <ul style="list-style-type: none"> <li>● <b>public</b>: balanceador de carga de red pública</li> <li>● <b>inner</b>: balanceador de carga de red privada</li> </ul> <p>Predeterminado: <b>inner</b></p>
bandwidth_name	Sí para los balanceadores de carga de red pública	String	<p>Nombre del ancho de banda. El valor predeterminado es <b>cce-bandwidth-*****</b>.</p> <p>Intervalo de valores: una string de 1 a 64 caracteres, incluidos letras minúsculas, dígitos y guiones bajos (_). El valor debe comenzar con una letra minúscula y terminar con una letra minúscula o un dígito.</p>

Parámetro	Obligatorio	Tipo	Descripción
bandwidth_chargemode	No	String	Modo de facturación de ancho de banda. <ul style="list-style-type: none"> <li>● <b>bandwidth</b>: facturado por ancho de banda</li> <li>● <b>traffic</b>: facturado por tráfico</li> </ul> Predeterminado: <b>bandwidth</b>
bandwidth_size	Sí para los balanceadores de carga de red pública	Integer	Tamaño del ancho de banda. El valor varía de 1 Mbit/s a 2000 Mbit/s de forma predeterminada. El rango real varía dependiendo de la configuración en cada región. <ul style="list-style-type: none"> <li>● El incremento mínimo para el ajuste de ancho de banda varía dependiendo del rango de ancho de banda. Los detalles son los siguientes:                             <ul style="list-style-type: none"> <li>– El incremento mínimo es de 1 Mbit/s si el ancho de banda permitido oscila entre 0 Mbit/s y 300 Mbit/s (con 300 Mbit/s incluidos).</li> <li>– El incremento mínimo es de 50 Mbit/s si el ancho de banda permitido varía de 300 Mbit/s a 1000 Mbit/s.</li> <li>– El incremento mínimo es de 500 Mbit/s si el ancho de banda permitido es mayor que 1000 Mbit/s.</li> </ul> </li> </ul>
bandwidth_sharetype	Sí para los balanceadores de carga de red pública	String	Tipo de ancho de banda. <b>PER</b> : ancho de banda dedicado.
eip_type	Sí para los balanceadores de carga de red pública	String	Tipo de la EIP. <ul style="list-style-type: none"> <li>● <b>5_telcom</b>: China Telecom</li> <li>● <b>5_union</b>: China Unicom</li> <li>● <b>5_bgp</b>: BGP dinámico</li> <li>● <b>5_sbgp</b>: BGP estático</li> </ul>

Parámetro	Obligatorio	Tipo	Descripción
name	No	String	<p>Nombre del balanceador de carga creado automáticamente.</p> <p>Intervalo de valores: una string de 1 a 64 caracteres, incluidos letras minúsculas, dígitos y guiones bajos (_). El valor debe comenzar con una letra minúscula y terminar con una letra minúscula o un dígito.</p> <p>Predeterminado: <b>cce-lb+ingress.UID</b></p>
vip_subnet_cidr_id	No	String	<p>Subred donde se encuentra el balanceador de carga. Este campo es compatible con clústeres de v1.21 o posterior.</p> <p>Si no se especifica este parámetro, el balanceador de carga y el clúster están en la misma subred.</p>
available_zone	Sí	Array of strings	<p>(Obligatorio) La AZ donde se encuentra el balanceador de carga.</p> <p>Puede obtener todas las AZ soportadas por <a href="#">consultar la lista de AZ</a>.</p> <p>Este parámetro solo está disponible para los balanceadores de carga dedicados.</p>
l4_flavor_name	No	String	<p>Nombre de la variante del balanceador de carga de capa 4.</p> <p>Puede obtener todos los tipos admitidos <a href="#">consultando la lista de variantes</a>.</p> <p>Este parámetro solo está disponible para los balanceadores de carga dedicados. El valor de este parámetro debe ser el mismo que el de <b>l7_flavor_name</b>, es decir, ambas son especificaciones elásticas o especificaciones fijas.</p>
l7_flavor_name	Sí	String	<p>(Obligatorio) El nombre de la variante del balanceador de carga de capa-7.</p> <p>Puede obtener todos los tipos admitidos <a href="#">consultando la lista de variantes</a>.</p> <p>Este parámetro solo está disponible para los balanceadores de carga dedicados.</p>

Parámetro	Obligatorio	Tipo	Descripción
elb_virsubnet_ids	No	Array of strings	<p>Subred donde se encuentra el servidor de backend del balanceador de carga. Si este parámetro se deja en blanco, se utiliza la subred de clúster predeterminada. Los balanceadores de carga ocupan un número diferente de direcciones IP de subred según sus especificaciones. Por lo tanto, no se recomienda utilizar los bloques CIDR de subred de otros recursos (como clústeres y nodos) como el bloque CIDR del balanceador de carga.</p> <p>Valor predeterminado: subred donde se encuentra el clúster</p> <p>Este parámetro solo está disponible para los balanceadores de carga dedicados.</p> <p>Ejemplo:</p> <pre>"elb_virsubnet_ids": [   "14567f27-8ae4-42b8-ae47-9f847a4690dd" ]</pre>

**Paso 3** Cree una entrada.

**kubectl create -f ingress-test.yaml**

Si se muestra la información similar a la siguiente, se ha creado la entrada.

```
ingress/ingress-test created
```

**kubectl get ingress**

Si se muestra información similar a la siguiente, la entrada se ha creado correctamente y se puede acceder a la carga de trabajo.

NAME	HOSTS	ADDRESS	PORTS	AGE
ingress-test	*	121.**.**.**	80	10s

**Paso 4** Ingrese **http://121.\*\*.\*\*.\*\*:80** en el cuadro de dirección del navegador para acceder a la carga de trabajo (por ejemplo, [Carga de trabajo de Nginx](#)).

**121.\*\*.\*\*.\*\*** indica la dirección IP del balanceador de carga unificado.

----Fin

## Creación de un ingreso - interconexión con un balanceador de carga existente

CCE le permite conectarse a un balanceador de carga existente al crear una entrada.

### NOTA

- Existing dedicated load balancers must be the application type (HTTP/HTTPS) supporting private networks (with a private IP).

**Si la versión del clúster es 1.23 o posterior, la configuración del archivo YAML es la siguiente:**

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress-test
  annotations:
    kubernetes.io/elb.id: <your_elb_id> # Replace it with the ID of your
existing load balancer.
    kubernetes.io/elb.ip: <your_elb_ip> # Replace it with your existing load
balancer IP.
    kubernetes.io/elb.port: '80'
spec:
  rules:
  - host: ''
    http:
      paths:
      - path: '/'
        backend:
          service:
            name: <your_service_name> # Replace it with the name of your target
Service.
            port:
              number: 8080 # Replace 8080 with your target service
port number.
          property:
            ingress.beta.kubernetes.io/url-match-mode: STARTS_WITH
            pathType: ImplementationSpecific
ingressClassName: cce
```

**Si la versión del clúster es 1.21 o anterior, la configuración del archivo YAML es la siguiente:**

```
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: ingress-test
  annotations:
    kubernetes.io/elb.id: <your_elb_id> # Replace it with the ID of your
existing load balancer.
    kubernetes.io/elb.ip: <your_elb_ip> # Replace it with your existing load
balancer IP.
    kubernetes.io/elb.port: '80'
    kubernetes.io/ingress.class: cce
spec:
  rules:
  - host: ''
    http:
      paths:
      - path: '/'
        backend:
          serviceName: <your_service_name> # Replace it with the name of your
target Service.
          servicePort: 80
          property:
            ingress.beta.kubernetes.io/url-match-mode: STARTS_WITH
```

**Tabla 7-26** Parámetros de clave

Parámetro	Obligatorio	Tipo	Descripción
kubernetes.io/elb.id	Sí	String	Este parámetro indica el ID de un balanceador de carga. El valor puede contener de 1 a 100 caracteres. <b>Cómo obtenerlo:</b> En la consola de gestión, haga clic en <b>Service List</b> y elija <b>Networking &gt; Elastic Load Balance</b> . Haga clic en el nombre del balanceador de carga de destino. En la página de ficha <b>Summary</b> , encuentre y copie el ID.
kubernetes.io/elb.ip	Sí	String	Este parámetro indica la dirección de servicio de un balanceador de carga. El valor puede ser la dirección IP pública de un balanceador de carga de red pública o la dirección IP privada de un balanceador de carga de red privada.

### 7.4.2.3 Configuración de certificados de HTTPS para ingresos de ELB

El ingreso admite la configuración de certificados de TLS y protege sus servicios con HTTPS.

Actualmente, puede utilizar el certificado de clave de TLS configurado en el clúster y el certificado de ELB.

 **NOTA**

Si HTTPS está habilitado para el mismo puerto del mismo balanceador de carga de múltiples entradas, debe seleccionar el mismo certificado.

### Uso de un certificado de clave de TLS

- Paso 1** Utilice `kubectl` para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).
- Paso 2** Ingress soporta dos tipos de claves TLS: `kubernetes.io/tls` y `IngressTLS`. Se usa como ejemplo `IngressTLS`. Para obtener más información, véase [Creación de un secreto](#).

Ejecute el siguiente comando para crear un archivo YAML llamado `ingress-test-secret.yaml` (el nombre del archivo se puede personalizar):

**vi ingress-test-secret.yaml**

**El archivo YAML se configura de la siguiente manera:**

```
apiVersion: v1
data:
  tls.crt: LS0*****tLS0tCg==
  tls.key: LS0tL*****0tLS0K
kind: Secret
metadata:
```

```

    annotations:
      description: test for ingressTLS secrets
      name: ingress-test-secret
      namespace: default
type: IngressTLS
    
```

 **NOTA**

En la información anterior, las **tls.crt** y **tls.key** solo son ejemplos. Reemplácelos con los archivos reales. Los valores de **tls.crt** y **tls.key** están codificados en Base64.

**Paso 3** Cree un secreto.

**kubectl create -f ingress-test-secret.yaml**

Si se muestra información similar a la siguiente, se está creando el secreto:

```
secret/ingress-test-secret created
```

Vea el secreto creado.

**kubectl get secrets**

Si se muestra la información similar a la siguiente, se ha creado el secreto:

NAME	TYPE	DATA	AGE
ingress-test-secret	IngressTLS	2	13s

**Paso 4** Cree un archivo YAML denominado **ingress-test.yaml**. El nombre del archivo se puede personalizar.

**vi ingress-test.yaml**

 **NOTA**

La política de seguridad predeterminada (kubernetes.io/elb.tls-ciphers-policy) solo se admite en clústeres de v1.17.17 o posterior.

La política de seguridad personalizada (kubernetes.io/elb.security\_policy\_id) solo se admite en clústeres de v1.17.17 o posterior.

**A continuación se utiliza el balanceador de carga creado automáticamente como ejemplo. El archivo YAML se configura de la siguiente manera:**

**Para clústeres de v1.21 o anterior:**

```

apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: ingress-test
  annotations:
    kubernetes.io/elb.class: union
    kubernetes.io/ingress.class: cce
    kubernetes.io/elb.port: '443'
    kubernetes.io/elb.autocreate:
      '{
        "type": "public",
        "bandwidth_name": "cce-bandwidth-15511633796**",
        "bandwidth_chargemode": "bandwidth",
        "bandwidth_size": 5,
        "bandwidth_sharetype": "PER",
        "eip_type": "5_bgp"
      }'
    kubernetes.io/elb.security_policy_id: 99bec42b-0dd4-4583-98e9-b05ce628d157 #
    The priority of the custom security policy is higher than that of the default
    security policy.
    kubernetes.io/elb.tls-ciphers-policy: tls-1-2
    
```

```
spec:
  tls:
  - secretName: ingress-test-secret
  rules:
  - host: ''
    http:
      paths:
      - path: '/'
        backend:
          serviceName: <your_service_name> # Replace it with the name of your
target Service.
          servicePort: 80
        property:
          ingress.beta.kubernetes.io/url-match-mode: STARTS_WITH
```

### Para clústeres de v1.23 o posterior:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress-test
  annotations:
    kubernetes.io/elb.class: union
    kubernetes.io/elb.port: '443'
    kubernetes.io/elb.autocreate:
      '{
        "type": "public",
        "bandwidth_name": "cce-bandwidth-15511633796**",
        "bandwidth_chargemode": "bandwidth",
        "bandwidth_size": 5,
        "bandwidth_sharetype": "PER",
        "eip_type": "5_bgp"
      }'
    kubernetes.io/elb.security_policy_id: 99bec42b-0dd4-4583-98e9-b05ce628d157 #
The priority of the custom security policy is higher than that of the default
security policy.
    kubernetes.io/elb.tls-ciphers-policy: tls-1-2
spec:
  tls:
  - secretName: ingress-test-secret
  rules:
  - host: ''
    http:
      paths:
      - path: '/'
        backend:
          service:
            name: <your_service_name> # Replace it with the name of your target
Service.
            port:
              number: 8080 # Replace 8080 with the port number of
your target Service.
          property:
            ingress.beta.kubernetes.io/url-match-mode: STARTS_WITH
            pathType: ImplementationSpecific
            ingressClassName: cce
```

**Tabla 7-27** Parámetros clave

Parámetro	Obligatorio	Tipo	Descripción
kubernetes.io/elb.security_policy_id	No	String	ID de la política de grupo de seguridad personalizada en ELB. Obténgalo en la consola de ELB. Este campo solo tiene efecto cuando se utiliza HTTPS y tiene una prioridad más alta que la política de seguridad predeterminada.  Para obtener más información acerca de cómo crear y actualizar una política de seguridad personalizada, consulte <a href="#">Política de seguridad de TLS</a> .
kubernetes.io/elb.tls-ciphers-policy	No	String	El valor predeterminado es <b>tls-1-2</b> , que es la política de seguridad predeterminada utilizada por el oyente y solo tiene efecto cuando se usa HTTPS.  Opciones: <ul style="list-style-type: none"> <li>● tls-1-0</li> <li>● tls-1-1</li> <li>● tls-1-2</li> <li>● tls-1-2-strict</li> </ul> Para obtener más información sobre los conjuntos de cifrado para cada política de seguridad, consulte <a href="#">Tabla 7-28</a> .
tls	No	Array of strings	Cuando se utiliza HTTPS, este parámetro debe agregarse para especificar el certificado de clave.  Se pueden agregar varios nombres de dominio y certificados independientes. Para obtener más información, véase <a href="#">Configuración de la Server Name Indication (SNI) para las entradas de ELB</a> .
secretName	No	String	Este parámetro es obligatorio si se utiliza HTTPS. Establezca este parámetro en el nombre del secreto creado.

**Tabla 7-28** Descripción de parámetro `tls_ciphers_policy`

Política de seguridad	Versión de TLS	Suite de cifrado
tls-1-0	TLS 1.2 TLS 1.1 TLS 1.0	ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:AES128-GCM-SHA256:AES256-GCM-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256:AES128-SHA256:AES256-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES128-SHA:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES256-SHA:AES128-SHA:AES256-SHA
tls-1-1	TLS 1.2 TLS 1.1	ECDHE-RSA-AES128-SHA256:AES128-GCM-SHA256:AES128-SHA256:AES256-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES128-SHA:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES256-SHA:AES128-SHA:AES256-SHA
tls-1-2	TLS 1.2	ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:AES128-GCM-SHA256:AES256-GCM-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256:AES128-SHA256:AES256-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384
tls-1-2-strict	TLS 1.2	ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:AES128-GCM-SHA256:AES256-GCM-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256:AES128-SHA256:AES256-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384

**Paso 5** Cree una entrada.

**kubectl create -f ingress-test.yaml**

Si se muestra la información similar a la siguiente, se ha creado la entrada.

```
ingress/ingress-test created
```

Vea la entrada creada.

**kubectl get ingress**

Si se muestra la información similar a la siguiente, se ha creado la entrada y se puede acceder a la carga de trabajo.

```
NAME          HOSTS          ADDRESS          PORTS          AGE
ingress-test  *              121.**.**.**       80             10s
```

**Paso 6** Introduzca **https://121.\*\*.\*\*.\*\*:443** en el cuadro de dirección del navegador para acceder a la carga de trabajo (por ejemplo, [Carga de trabajo de Nginx](#)).

**121.\*\*.\*\*.\*\*** indica la dirección IP del balanceador de carga unificado.

----**Fin**

## Uso del certificado de ELB

Para utilizar el certificado de ELB, puede especificar el `kubernetes.io/elb.tls-certificate-ids` de anotaciones.

### 📖 NOTA

1. Si especifica tanto el certificado IngressTLS como el certificado de ELB, se utiliza este último.
2. CCE no comprueba si el certificado de ELB es válido. Solo comprueba si el certificado existe.

```
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: ingress-test
  annotations:
    kubernetes.io/ingress.class: cce
    kubernetes.io/elb.port: '443'
    kubernetes.io/elb.id: 0b9a6c4d-bd8b-45cc-bfc8-ff0f9da54e95
    kubernetes.io/elb.class: union
    kubernetes.io/elb.tls-certificate-ids:
058cc023690d48a3867ad69dbe9cd6e5,b98382b1f01c473286653afd1ed9ab63
spec:
  rules:
  - host: ''
    http:
      paths:
      - path: '/'
        backend:
          serviceName: <your_service_name> # Replace it with the name of your
target Service.
          servicePort: 80
        property:
          ingress.beta.kubernetes.io/url-match-mode: STARTS_WITH
```

### 7.4.2.4 Configuración de la Server Name Indication (SNI) para las entradas de ELB

SNI permite que se proporcionen múltiples nombres de dominio de acceso basados en TLS para sistemas externos que utilicen la misma dirección IP y número de puerto. Diferentes nombres de dominio pueden utilizar diferentes certificados de seguridad.

### 📖 NOTA

- Solo se puede especificar un nombre de dominio para cada certificado de SNI. Se admiten los certificados de dominio comodín.
- La política de seguridad (`kubernetes.io/elb.tls-ciphers-policy`) solo se admite en clústeres de v1.17.11 o posterior.

Puede habilitar SNI cuando se cumplan las condiciones anteriores. A continuación se utiliza la creación automática de un balanceador de carga como ejemplo. En este ejemplo, `sni-test-secret-1` y `sni-test-secret-2` son los certificados de SNI. Los nombres de dominio especificados por los certificados deben ser los mismos que los de los certificados.

#### Para clústeres de v1.21 o anterior:

```
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: ingress-test
  annotations:
    kubernetes.io/elb.class: union
    kubernetes.io/ingress.class: cce
    kubernetes.io/elb.port: '443'
    kubernetes.io/elb.autocreate:
```

```

        '{
          "type": "public",
          "bandwidth_name": "cce-bandwidth-*****",
          "bandwidth_chargemode": "bandwidth",
          "bandwidth_size": 5,
          "bandwidth_sharetype": "PER",
          "eip_type": "5_bgp"
        }'
      kubernetes.io/elb.tls-ciphers-policy: tls-1-2
spec:
  tls:
  - secretName: ingress-test-secret
  - hosts:
    - example.top # Domain name specified when a certificate is issued
      secretName: sni-test-secret-1
  - hosts:
    - example.com # Domain name specified when a certificate is issued
      secretName: sni-test-secret-2
  rules:
  - host: ''
    http:
      paths:
      - path: '/'
        backend:
          serviceName: <your_service_name> # Replace it with the name of your
target Service.
          servicePort: 80
        property:
          ingress.beta.kubernetes.io/url-match-mode: STARTS_WITH

```

### Para clústeres de v1.23 o posterior:

```

apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress-test
  annotations:
    kubernetes.io/elb.class: union
    kubernetes.io/elb.port: '443'
    kubernetes.io/elb.autocreate:
      '{
        "type": "public",
        "bandwidth_name": "cce-bandwidth-*****",
        "bandwidth_chargemode": "bandwidth",
        "bandwidth_size": 5,
        "bandwidth_sharetype": "PER",
        "eip_type": "5_bgp"
      }'
    kubernetes.io/elb.tls-ciphers-policy: tls-1-2
spec:
  tls:
  - secretName: ingress-test-secret
  - hosts:
    - example.top # Domain name specified when a certificate is issued
      secretName: sni-test-secret-1
  - hosts:
    - example.com # Domain name specified when a certificate is issued
      secretName: sni-test-secret-2
  rules:
  - host: ''
    http:
      paths:
      - path: '/'
        backend:
          service:
            name: <your_service_name> # Replace it with the name of your target
Service.
            port:
              number: 8080 # Replace 8080 with the port number of
your target Service.
        property:

```

```
ingress.beta.kubernetes.io/url-match-mode: STARTS_WITH
pathType: ImplementationSpecific
ingressClassName: cce
```

### 7.4.2.5 Ingreso de ELB encamina a varios servicios

Los ingresos pueden encaminar a varios servicios de backend según diferentes políticas coincidentes. El campo **spec** en el archivo YAML se establece como se muestra a continuación. Puede acceder a **www.example.com/foo**, **www.example.com/bar** y **foo.example.com/** para encaminar a tres servicios de backend diferentes.

#### AVISO

El URL registrado en una política de reenvío de ingreso debe ser la misma que la dirección URL utilizada para acceder al Service de backend. De lo contrario, se devolverá un error 404.

```
spec:
  rules:
  - host: 'www.example.com'
    http:
      paths:
      - path: '/foo'
        backend:
          serviceName: <your_service_name> # Replace it with the name of your
          target Service.
          servicePort: 80
        property:
          ingress.beta.kubernetes.io/url-match-mode: STARTS_WITH
      - path: '/bar'
        backend:
          serviceName: <your_service_name> # Replace it with the name of your
          target Service.
          servicePort: 80
        property:
          ingress.beta.kubernetes.io/url-match-mode: STARTS_WITH
  - host: 'foo.example.com'
    http:
      paths:
      - path: '/'
        backend:
          serviceName: <your_service_name> # Replace it with the name of your
          target Service.
          servicePort: 80
        property:
          ingress.beta.kubernetes.io/url-match-mode: STARTS_WITH
```

### 7.4.2.6 Ingresos de ELB usando HTTP/2

Los ingresos pueden usar HTTP/2 para exponer los servicios. Las conexiones desde el balanceador de carga a su aplicación usan HTTP/1.X de forma predeterminada. Si su aplicación es capaz de recibir solicitudes de HTTP2, puede agregar el siguiente campo a la anotación de ingreso para habilitar el uso de HTTP/2:

```
kubernetes.io/elb.http2-enable: 'true'
```

A continuación se muestra el archivo YAML para asociarse con un balanceador de carga existente:

**Para clústeres de v1.21 o anterior:**

```
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
```

```

metadata:
  name: ingress-test
  annotations:
    kubernetes.io/elb.id: <your_elb_id> # Replace it with the ID of your
existing load balancer.
    kubernetes.io/elb.ip: <your_elb_ip> # Replace it with the IP of your
existing load balancer.
    kubernetes.io/elb.port: '443'
    kubernetes.io/ingress.class: cce
    kubernetes.io/elb.http2-enable: 'true' # Enable HTTP/2.
spec:
  tls:
  - secretName: ingress-test-secret
  rules:
  - host: ''
    http:
      paths:
      - path: '/'
        backend:
          serviceName: <your_service_name> # Replace it with the name of your
target Service.
          servicePort: 80 # Replace it with the port number of
your target Service.
        property:
          ingress.beta.kubernetes.io/url-match-mode: STARTS_WITH
    
```

**Para clústeres de v1.23 o posterior:**

```

apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress-test
  annotations:
    kubernetes.io/elb.id: <your_elb_id> # Replace it with the ID of your
existing load balancer.
    kubernetes.io/elb.ip: <your_elb_ip> # Replace it with the IP of your
existing load balancer.
    kubernetes.io/elb.port: '443'
    kubernetes.io/elb.http2-enable: 'true' # Enable HTTP/2.
spec:
  tls:
  - secretName: ingress-test-secret
  rules:
  - host: ''
    http:
      paths:
      - path: '/'
        backend:
          service:
            name: <your_service_name> # Replace it with the name of your target
Service.
            port:
              number: 8080 # Replace 8080 with the port number of
your target Service.
          property:
            ingress.beta.kubernetes.io/url-match-mode: STARTS_WITH
            pathType: ImplementationSpecific
        ingressClassName: cce
    
```

Tabla 6 Parámetros de HTTP/2

Parámetro	Obligatorio	Tipo	Descripción
kubernetes.io/elb.http2-enable	No	Bool	<p>Si HTTP/2 está habilitado. El reenvío de solicitudes mediante HTTP/2 mejora el rendimiento de acceso entre su aplicación y el balanceador de carga. Sin embargo, el balanceador de carga todavía usa HTTP 1.X para reenviar solicitudes al servidor backend. <b>Este parámetro es compatible con clústeres de v1.19.16-r0, v1.21.3-r0 y versiones posteriores.</b></p> <p>Opciones:</p> <ul style="list-style-type: none"> <li>● <b>true</b>: activado</li> <li>● <b>false</b>: deshabilitado (valor predeterminado)</li> </ul> <p>Nota: <b>HTTP/2 se puede habilitar o deshabilitar solo cuando el oyente usa HTTPS.</b> Este parámetro no es válido cuando el protocolo de oyente es HTTP y el valor predeterminado es <b>false</b>.</p>

### 7.4.2.7 Configuración de ingresos de ELB con anotaciones

Al agregar anotaciones a un archivo YAML, puede implementar funciones de ingreso más avanzadas. En esta sección se describen las anotaciones que se pueden utilizar al crear una entrada del tipo ELB.

## Interconexión con ELB

Tabla 7-29 Anotaciones de interconexión con ELB

Parámetro	Tipo	Descripción	Versión de clúster admitida
kubernetes.io/elb.class	String	<p>Seleccione un tipo de balanceador de carga adecuado.</p> <p>El valor puede ser:</p> <ul style="list-style-type: none"> <li>● <b>union</b>: balanceador de carga compartido</li> <li>● <b>performance</b>: balanceador de carga dedicado, que solo se puede utilizar en clústeres de v1.17 y posteriores. Para obtener más información, consulte <a href="#">Diferencias entre los balanceadores de carga compartidos y los dedicados</a>.</li> </ul>	v1.9 o posterior

Parámetro	Tipo	Descripción	Versión de clúster admitida
kubernetes.io/ingress.class	String	<ul style="list-style-type: none"> <li>● <b>cce</b>: Se utiliza la entrada de ELB autodesarrollada.</li> <li>● <b>nginx</b>: Se utiliza la entrada de Nginx.</li> </ul> Este parámetro es obligatorio cuando se crea una entrada llamando a la API. Para los clústeres de v1.23 o posterior, utilice el parámetro <b>ingressClassName</b> . Para obtener más información, véase <a href="#">Uso de kubectl para crear una entrada de ELB</a> .	Solo clústeres de v1.21 o anteriores
kubernetes.io/elb.port	Integer	Este parámetro indica el puerto externo registrado con la dirección del LoadBalancer Service. Rango soportado: 1 a 65535	v1.9 o posterior
kubernetes.io/elb.id	String	Obligatorio <b>cuando se va a interconectar un balanceador de carga existente</b> . ID de un balanceador de carga. <b>Cómo obtenerlo:</b> En la consola de gestión, haga clic en <b>Service List</b> y elija <b>Networking &gt; Elastic Load Balance</b> . Haga clic en el nombre del balanceador de carga de destino. En la página de ficha <b>Summary</b> , encuentre y copie el ID.	v1.9 o posterior
kubernetes.io/elb.ip	String	Obligatorio <b>cuando se va a interconectar un balanceador de carga existente</b> . Este parámetro indica la dirección de servicio de un balanceador de carga. El valor puede ser la dirección IP pública de un balanceador de carga de red pública o la dirección IP privada de un balanceador de carga de red privada.	v1.9 o posterior

Parámetro	Tipo	Descripción	Versión de clúster admitida
kubernetes.io/ elb.autocreate	<b>Tabla 7-31</b> Object	<p>Obligatorio <b>cuando los balanceadores de carga se crean automáticamente.</b></p> <p><b>Ejemplo</b></p> <ul style="list-style-type: none"> <li>● Si se creará automáticamente un balanceador de carga de red pública, establezca este parámetro en el siguiente valor:                      '{"type":"public","bandwidth_name":"cce-bandwidth-1551163379627","bandwidth_chargemode":"bandwidth","bandwidth_size":5,"bandwidth_sharetype":"PER","eip_type":"5_bgp","name":"james"}'</li> <li>● Si se creará automáticamente un balanceador de carga de red privada, establezca este parámetro en el siguiente valor:                      '{"type":"inner","name":"A-location-d-test"}'</li> </ul>	v1.9 o posterior
kubernetes.io/ elb.enterpriseID	String	<p>Opcional <b>cuando los balanceadores de carga se crean automáticamente.</b></p> <p><b>Los clústeres de v1.15 y posteriores admiten este campo. En los clústeres anteriores a v1.15, los balanceadores de carga se crean en el proyecto predeterminado de forma predeterminada.</b></p> <p>Este parámetro indica el ID del proyecto de empresa en el que se creará el balanceador de carga de ELB.</p> <p>Si este parámetro no se especifica o se establece en <b>0</b>, los recursos estarán enlazados al proyecto de empresa predeterminado.</p> <p><b>Cómo obtenerlo:</b></p> <p>Inicie sesión en la consola de gestión y seleccione <b>Enterprise &gt; Project Management</b> en la barra de menú superior. En la lista que se muestra, haga clic en el nombre del proyecto de empresa de destino y copie el ID en la página de detalles del proyecto de empresa.</p>	v1.15 o posterior

Parámetro	Tipo	Descripción	Versión de clúster admitida
kubernetes.io/elb.subnet-id	String	<p>Opcional <b>cuando los balanceadores de carga se crean automáticamente.</b></p> <p>ID de la subred donde se encuentra el clúster. El valor puede contener de 1 a 100 caracteres.</p> <ul style="list-style-type: none"> <li>● Obligatorio cuando se va a crear automáticamente un clúster de v1.11.7-r0 o anterior.</li> <li>● Opcional para los clústeres posteriores a v1.11.7-r0.</li> </ul>	<p>Obligatorio para los clústeres anteriores a v1.11.7-r0</p> <p>Descartado en los clústeres posteriores a v1.11.7-r0</p>

Para utilizar las anotaciones anteriores, realice los pasos siguientes:

- Consulte [Creación de un ingreso - interconexión con un balanceador de carga existente](#) para interconectar un balanceador de carga existente.
- Consulte [Creación de una entrada - Creación automática de un balanceador de carga](#) para crear automáticamente un balanceador de carga.

## Uso de HTTP/2

**Tabla 7-30** Anotaciones de uso de HTTP/2

Parámetro	Tipo	Descripción	Versión de clúster admitida
kubernetes.io/elb.http2-enable	String	<p>Si HTTP/2 está habilitado. El reenvío de solicitudes mediante HTTP/2 mejora el rendimiento de acceso entre su aplicación y el balanceador de carga. Sin embargo, el balanceador de carga todavía usa HTTP 1.X para reenviar solicitudes al servidor backend. <b>Este parámetro es compatible con los clústeres de v1.19.16-r0, v1.21.3-r0 y posteriores.</b></p> <p>Opciones:</p> <ul style="list-style-type: none"> <li>● <b>true</b>: activado</li> <li>● <b>false</b>: deshabilitado (valor predeterminado)</li> </ul> <p>Nota: <b>HTTP/2 se puede habilitar o deshabilitar solo cuando el oyente usa HTTPS.</b> Este parámetro no es válido y por defecto es <b>false</b> cuando el protocolo de oyente es HTTP.</p>	v1.19.16-r0, v1.21.3-r0 o posterior

Para obtener más información sobre los escenarios de la aplicación, consulte [Ingresos de ELB usando HTTP/2](#).

## Estructura de datos

**Tabla 7-31** Estructura de datos del campo elb.autocreate

Parámetro	Obligatorio	Tipo	Descripción
type	No	String	<p>Tipo de red del balanceador de carga.</p> <ul style="list-style-type: none"> <li>● <b>public</b>: balanceador de carga de red pública</li> <li>● <b>inner</b>: balanceador de carga de red privada</li> </ul> <p>Predeterminado: <b>inner</b></p>

Parámetro	Obligatorio	Tipo	Descripción
bandwidth_name	Sí para los balanceadores de carga de red pública	String	<p>Nombre del ancho de banda. El valor predeterminado es <b>cce-bandwidth-*****</b>.</p> <p>Intervalo de valores: una string de 1 a 64 caracteres, incluidos letras minúsculas, dígitos y guiones bajos (_). El valor debe comenzar con una letra minúscula y terminar con una letra minúscula o un dígito.</p>
bandwidth_chargemode	No	String	<p>Modo de facturación de ancho de banda.</p> <ul style="list-style-type: none"> <li>● <b>bandwidth</b>: facturado por ancho de banda</li> <li>● <b>traffic</b>: facturado por tráfico</li> </ul> <p>Predeterminado: <b>bandwidth</b></p>
bandwidth_size	Sí para los balanceadores de carga de red pública	Integer	<p>Tamaño del ancho de banda. El valor varía de 1 Mbit/s a 2000 Mbit/s de forma predeterminada. El rango real varía dependiendo de la configuración en cada región.</p> <ul style="list-style-type: none"> <li>● El incremento mínimo para el ajuste de ancho de banda varía dependiendo del rango de ancho de banda. Los detalles son los siguientes:                             <ul style="list-style-type: none"> <li>– El incremento mínimo es de 1 Mbit/s si el ancho de banda permitido oscila entre 0 Mbit/s y 300 Mbit/s (con 300 Mbit/s incluidos).</li> <li>– El incremento mínimo es de 50 Mbit/s si el ancho de banda permitido varía de 300 Mbit/s a 1000 Mbit/s.</li> <li>– El incremento mínimo es de 500 Mbit/s si el ancho de banda permitido es mayor que 1000 Mbit/s.</li> </ul> </li> </ul>
bandwidth_sharetype	Sí para los balanceadores de carga de red pública	String	<p>Tipo de ancho de banda.</p> <p><b>PER</b>: ancho de banda dedicado.</p>

Parámetro	Obligatorio	Tipo	Descripción
eip_type	Sí para los balanceadores de carga de red pública	String	Tipo de la EIP. <ul style="list-style-type: none"> <li>● <b>5_telcom</b>: China Telecom</li> <li>● <b>5_union</b>: China Unicom</li> <li>● <b>5_bgp</b>: BGP dinámico</li> <li>● <b>5_sbgp</b>: BGP estático</li> </ul>
name	No	String	Nombre del balanceador de carga creado automáticamente. Intervalo de valores: una string de 1 a 64 caracteres, incluidos letras minúsculas, dígitos y guiones bajos (_). El valor debe comenzar con una letra minúscula y terminar con una letra minúscula o un dígito. Predeterminado: <b>cce-lb+ingress.UID</b>
vip_subnet_cidr_id	No	String	Subred donde se encuentra el balanceador de carga. Este campo es compatible con clústeres de v1.21 o posterior. Si no se especifica este parámetro, el balanceador de carga y el clúster están en la misma subred.
available_zone	Sí	Array of strings	(Obligatorio) La AZ donde se encuentra el balanceador de carga. Puede obtener todas las AZ soportadas por <a href="#">consultar la lista de AZ</a> . Este parámetro solo está disponible para los balanceadores de carga dedicados.
l4_flavor_name	No	String	Nombre de la variante del balanceador de carga de capa 4. Puede obtener todos los tipos admitidos <a href="#">consultando la lista de variantes</a> . Este parámetro solo está disponible para los balanceadores de carga dedicados. El valor de este parámetro debe ser el mismo que el de <b>l7_flavor_name</b> , es decir, ambas son especificaciones elásticas o especificaciones fijas.

Parámetro	Obligatorio	Tipo	Descripción
l7_flavor_name	Sí	String	(Obligatorio) El nombre de la variante del balanceador de carga de capa-7. Puede obtener todos los tipos admitidos <a href="#">consultando la lista de variantes</a> . Este parámetro solo está disponible para los balanceadores de carga dedicados.
elb_virsubnet_ids	No	Array of strings	Subred donde se encuentra el servidor de backend del balanceador de carga. Si este parámetro se deja en blanco, se utiliza la subred de clúster predeterminada. Los balanceadores de carga ocupan un número diferente de direcciones IP de subred según sus especificaciones. Por lo tanto, no se recomienda utilizar los bloques CIDR de subred de otros recursos (como clústeres y nodos) como el bloque CIDR del balanceador de carga. Valor predeterminado: subred donde se encuentra el clúster Este parámetro solo está disponible para los balanceadores de carga dedicados. Ejemplo: <pre>"elb_virsubnet_ids": [   "14567f27-8ae4-42b8-ae47-9f847a4690dd" ]</pre>

## 7.4.3 Ingresos de Nginx

### 7.4.3.1 Creación de entradas de Nginx en la consola

#### Requisitos previos

- Un ingreso proporciona acceso a la red para cargas de trabajo back-end. Asegúrese de que una carga de trabajo esté disponible en un clúster. Si no hay ninguna carga de trabajo disponible, despliegue una carga de trabajo haciendo referencia a [Creación de una Deployment](#), [Creación de un StatefulSet](#) o [Creación de un DaemonSet](#).
- Si necesita agregar una entrada de Nginx, asegúrese de que el complemento **nginx-ingress** se haya instalado en el clúster. Para obtener más información, véase [Instalación del complemento](#).

## Precauciones

- **No se recomienda modificar ninguna configuración de un balanceador de carga en la consola de ELB. De lo contrario, el Service será anormal.** Si ha modificado la configuración, desinstale el complemento nginx-ingress y vuelva a instalarlo.
- El URL registrado en una política de reenvío de ingreso debe ser la misma que la dirección URL utilizada para acceder al Service de backend. De lo contrario, se devolverá un error 404.
- El balanceador de carga seleccionado o creado debe estar en la misma VPC que el clúster actual y debe coincidir con el tipo de balanceador de carga (la red privada o la pública).
- El balanceador de carga tiene al menos dos oyentes, y los puertos 80 y 443 no están ocupados por oyentes.

## Creación de un ingreso de Nginx

Esta sección utiliza una carga de trabajo de Nginx como ejemplo para describir cómo crear una entrada de Nginx.

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** Elija **Networking** en el panel de navegación, haga clic en la ficha **Ingresses** y haga clic en **Create Service** en la esquina superior derecha.

**Paso 3** Establezca los parámetros de entrada.

- **Name:** Especifique el nombre de una entrada, por ejemplo, **nginx-ingress-demo**.
- **Namespace:** Seleccione el espacio de nombres al que se va a agregar la entrada.
- **nginx-ingress:** Esta opción solo se muestra cuando el complemento de **nginx-ingress** se ha instalado en el clúster.

Después de activar , se interconecta nginx-ingress para proporcionar acceso a la capa 7. Puede configurar los siguientes parámetros:

**TLS:** nginx-ingress soporta HTTP y HTTPS. El puerto de escucha predeterminado reservado durante la instalación de nginx-ingress es **80** para las solicitudes HTTP y **443** para las solicitudes de HTTPS. Para usar HTTPS, debe configurar el certificado de servidor.

- **Server Certificate:** Al crear un oyente de HTTPS, debe vincular un certificado de TLS para admitir la autenticación cifrada para la transmisión de datos de HTTPS. Para obtener más información sobre cómo crear un secreto, consulte [Creación de un secreto](#).
- **SNI:** Server Name Indication (SNI) es un protocolo extendido de TLS. Permite proporcionar múltiples nombres de dominio de acceso basados en TLS para sistemas externos que utilizan la misma dirección IP y puerto. Diferentes nombres de dominio pueden utilizar diferentes certificados de seguridad. Después de habilitar el SNI, el cliente puede enviar el nombre de dominio solicitado al iniciar una solicitud de handshake de TLS. Después de recibir la solicitud de TLS, el balanceador de carga busca el certificado basado en el nombre de dominio en la solicitud. Si se encuentra el certificado correspondiente al nombre de dominio, el balanceador de carga devuelve el certificado para la autorización. De lo contrario, se devuelve el certificado predeterminado (certificado de servidor) para la autorización.

- **Forwarding Policies:** Cuando la dirección de acceso de una solicitud coincide con la política de reenvío (una política de reenvío consiste en un nombre de dominio y un URL), la solicitud se reenvía al Service de destino correspondiente para su procesamiento. Haga clic en **Add Forwarding Policies** para agregar varias políticas de reenvío.
  - **Domain Name:** nombre de dominio real. Asegúrese de que el nombre de dominio introducido ha sido registrado y archivado. Después de crear la entrada, vincule el nombre de dominio a la dirección IP del balanceador de carga creado automáticamente (dirección IP de la dirección de acceso de entrada). Si se configura una regla de nombre de dominio, siempre se debe usar el nombre de dominio para obtener acceso.
  - **URL:** ruta de acceso a registrar, por ejemplo, **/healthz**.

#### **NOTA**

- La regla de coincidencia de ruta de acceso de entrada de Nginx se basa en el prefijo de ruta separado por la barra diagonal (/) y distingue entre mayúsculas y minúsculas. Si la ruta secundaria separada por una barra diagonal (/) coincide con el prefijo, el acceso es normal. Sin embargo, si el prefijo es solo una parte de la cadena de caracteres en la ruta secundaria, el acceso no coincide. Por ejemplo, si la dirección URL está establecida en /healthz, /healthz/v1 coincide, pero /healthzv1 no coincide.
- La ruta de acceso agregada aquí debe existir en la aplicación de backend. De lo contrario, el reenvío falla.

Por ejemplo, el URL de acceso predeterminado de la aplicación Nginx es **/usr/share/nginx/html**. Al agregar **/test** a la política de reenvío de ingreso, asegúrese de que su aplicación de Nginx contiene el mismo URL, es decir, **/usr/share/nginx/html/test**, de lo contrario, se devuelve 404.
- **Destination Service:** Seleccione un Service existente o cree un Service. Los Service que no cumplen los criterios de búsqueda se eliminan automáticamente.
- **Destination Service Port:** Seleccione el puerto de acceso del Service de destino.
- **Operation:** Haga clic en **Delete** para eliminar la configuración.
- **Annotation:** El valor tiene el formato key:value. Puede utilizar **anotaciones** para consultar las configuraciones admitidas por nginx-ingress.

**Paso 4** Una vez completada la configuración, haga clic en **Create**.

Después de crear la entrada, se muestra en la lista de entrada.

----Fin

## 7.4.3.2 Uso de kubectl para crear una entrada de Nginx

### Escenario

Esta sección utiliza una **carga de trabajo de Nginx** como ejemplo para describir cómo crear una entrada de Nginx con kubectl.

### Requisitos previos

- El complemento nginx-ingress se ha instalado en un clúster. Para obtener más información, véase **Instalación del complemento**.
- Un ingreso proporciona acceso a la red para cargas de trabajo backend. Asegúrese de que una carga de trabajo esté disponible en un clúster. Si no hay ninguna carga de trabajo

disponible, despliegue una carga de trabajo haciendo referencia a [Creación de una Deployment](#), [Creación de un StatefulSet](#) o [Creación de un DaemonSet](#).

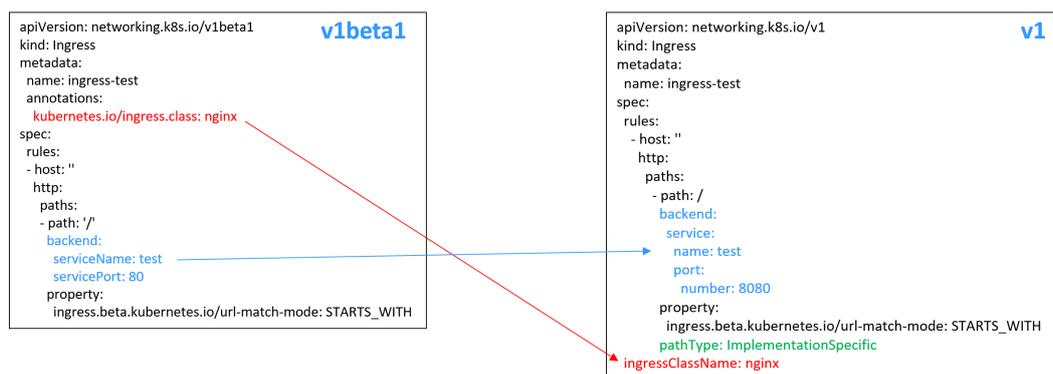
- Se ha configurado un Service ClusterIP o NodePort para la carga de trabajo. Para obtener más información acerca de cómo configurar el Service, consulte [ClusterIP](#) o [NodePort](#).
- Si se utiliza HTTPS para el acceso externo, debe crear un secreto de IngressTLS por adelantado. Para obtener más información sobre cómo crear un secreto, consulte [Creación de un secreto](#).

## Descripción de ingreso de networking.k8s.io/v1

En los clústeres de CCE de v1.23 o posterior, la versión de ingreso se cambia a **networking.k8s.io/v1**.

Comparado con v1beta1, v1 tiene las siguientes diferencias en parámetros:

- El tipo de entrada se cambia de **kubernetes.io/ingress.class** en **annotations** a **spec.ingressClassName**.
- Se cambia el formato de **backend**.
- El parámetro **pathType** debe especificarse para cada ruta. Las opciones son las siguientes:
  - **ImplementationSpecific**: El método de coincidencia depende del controlador de entrada. El método de coincidencia definido por **ingress.beta.kubernetes.io/url-match-mode** se usa en CCE, que es el mismo que v1beta1.
  - **Exact**: coincidencia exacta del URL, que distingue entre mayúsculas y minúsculas.
  - **Prefix**: coincidencia basada en el prefijo de URL separado por una barra diagonal (/). La coincidencia distingue entre mayúsculas y minúsculas, y los elementos de la ruta se hacen coincidir uno por uno. Un elemento de trazado hace referencia a una lista de etiquetas en el trazado separadas por una barra diagonal (/).



## Creación de un ingreso de Nginx

**Paso 1** Utilice kubectl para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).

**Paso 2** Cree un archivo YAML denominado **ingress-test.yaml**. El nombre del archivo se puede personalizar.

**vi ingress-test.yaml**

 **NOTA**

A partir del clúster v1.23, la versión de ingreso cambia de **networking.k8s.io/v1beta1** a **networking.k8s.io/v1**. Para obtener más información sobre las diferencias entre v1 y v1beta1, consulte [Descripción de ingreso de networking.k8s.io/v1](#).

**A continuación se utiliza HTTP como ejemplo para describir cómo configurar el archivo YAML:**

**Si el nodo está en un clúster de v1.23 o posterior:**

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress-test
spec:
  rules:
    - host: ''
      http:
        paths:
          - path: /
            backend:
              service:
                name: <your_service_name> # Replace it with the name of your
target Service.
            port:
              number: <your_service_port> # Replace it with the port number
of your target Service.
          property:
            ingress.beta.kubernetes.io/url-match-mode: STARTS_WITH
          pathType: ImplementationSpecific
ingressClassName: nginx # Nginx ingress is used.
```

**Si el nodo está en un clúster de v1.21 o anterior:**

```
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: ingress-test
  namespace: default
  annotations:
    kubernetes.io/ingress.class: nginx # Nginx ingress is used.
spec:
  rules:
    - host: ''
      http:
        paths:
          - path: '/'
            backend:
              serviceName: <your_service_name> # Replace it with the name of
your target Service.
              servicePort: <your_service_port> # Replace it with the port number
of your target Service.
```

**Tabla 7-32** Parámetros de clave

Parámetro	Obligato	Tipo	Descripción
kubernetes.io/ingress.class	Sí (solo para los clústeres de v1.21 o anteriores)	String	<b>nginx</b> : indica que se utiliza la entrada de Nginx. Esta opción no se puede utilizar si el complemento de nginx-ingress no está instalado.  Este parámetro es obligatorio cuando se crea una entrada invocando a la API.

Parámetro	Obligatorio	Tipo	Descripción
ingressClassName	Sí (solo para los clústeres de v1.23 o posterior)	String	<b>nginx</b> : indica que se utiliza la entrada de Nginx. Esta opción no se puede utilizar si el complemento de nginx-ingress no está instalado.  Este parámetro es obligatorio cuando se crea una entrada invocando a la API.
host	No	String	Nombre de dominio para acceder al Service. De forma predeterminada, este parámetro se deja en blanco y el nombre de dominio debe coincidir completamente. Asegúrese de que el nombre de dominio ha sido registrado y archivado. Una vez configurada una regla de nombre de dominio, debe usar el nombre de dominio para tener acceso.
path	Sí	String	Ruta de ruta definida por el usuario. Todas las solicitudes de acceso externo deben coincidir con <b>host</b> y <b>path</b> . <b>NOTA</b> <ul style="list-style-type: none"> <li>● La regla de coincidencia de ruta de acceso de entrada de Nginx se basa en el prefijo de ruta separado por la barra diagonal (/) y distingue entre mayúsculas y minúsculas. Si la ruta secundaria separada por una barra diagonal (/) coincide con el prefijo, el acceso es normal. Sin embargo, si el prefijo es solo una parte de la cadena de caracteres en la ruta secundaria, el acceso no coincide. Por ejemplo, si la dirección URL está establecida en /healthz, /healthz/v1 coincide, pero /healthzv1 no coincide.</li> <li>● La ruta de acceso agregada aquí debe existir en la aplicación de backend. De lo contrario, el reenvío falla. Por ejemplo, el URL de acceso predeterminado de la aplicación Nginx es <b>/usr/share/nginx/html</b>. Al agregar <b>/test</b> a la política de reenvío de ingreso, asegúrese de que su aplicación de Nginx contiene el mismo URL, es decir, <b>/usr/share/nginx/html/test</b>, de lo contrario, se devuelve 404.</li> </ul>

Parámetro	Obligatorio	Tipo	Descripción
ingress.beta.kubernetes.io/url-match-mode	No	String	Política de coincidencia de rutas. Predeterminado: <b>STARTS_WITH</b> (Coincidencia de prefijo) Opciones: <ul style="list-style-type: none"> <li>● <b>EQUAL_TO</b>: coincidencia exacta</li> <li>● <b>STARTS_WITH</b>: coincidencia de prefijos</li> <li>● <b>REGEX</b>: coincidencia de expresiones regulares</li> </ul>
pathType	Sí	String	Tipo de ruta. Este campo solo es compatible con los clústeres de v1.23 o posterior. <ul style="list-style-type: none"> <li>● <b>ImplementationSpecific</b>: El método de coincidencia depende del controlador de entrada. El método de emparejamiento definido por <b>ingress.beta.kubernetes.io/url-match-mode</b> se usa en CCE.</li> <li>● <b>Exact</b>: coincidencia exacta del URL, que distingue entre mayúsculas y minúsculas.</li> <li>● <b>Prefix</b>: coincidencia basada en el prefijo de URL separado por una barra diagonal (/). La coincidencia distingue entre mayúsculas y minúsculas, y los elementos de la ruta se hacen coincidir uno por uno. Un elemento de trazado hace referencia a una lista de etiquetas en el trazado separadas por una barra diagonal (/).</li> </ul>

**Paso 3** Cree una entrada.

**kubectl create -f ingress-test.yaml**

Si se muestra la información similar a la siguiente, se ha creado la entrada.

```
ingress/ingress-test created
```

Vea la entrada creada.

**kubectl get ingress**

Si se muestra información similar a la siguiente, la entrada se ha creado correctamente y se puede acceder a la carga de trabajo.

```
NAME          HOSTS          ADDRESS          PORTS          AGE
ingress-test  *             121.**.**.**      80             10s
```

**Paso 4** Ingrese **http://121.\*\*.\*\*.\*:80** en el cuadro de dirección del navegador para acceder a la carga de trabajo (por ejemplo, [Carga de trabajo de Nginx](#)).

**121.\*\*.\*\*.\*** indica la dirección IP del balanceador de carga unificado.

---Fin

### 7.4.3.3 Configuración de certificados de HTTPS para entradas de Nginx

Los certificados de HTTPS se pueden configurar para el ingreso para proporcionar servicios de seguridad.

**Paso 1** Utilice `kubectl` para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).

**Paso 2** Ingress soporta dos tipos de claves TLS: `kubernetes.io/tls` y `IngressTLS`. Se usa como ejemplo `IngressTLS`. Para obtener más información, véase [Creación de un secreto](#).

Ejecute el siguiente comando para crear un archivo YAML llamado **ingress-test-secret.yaml** (el nombre del archivo se puede personalizar):

**vi ingress-test-secret.yaml**

**El archivo YAML se configura de la siguiente manera:**

```
apiVersion: v1
data:
  tls.crt: LS0*****tLS0tCg==
  tls.key: LS0tL*****0tLS0K
kind: Secret
metadata:
  annotations:
    description: test for ingressTLS secrets
    name: ingress-test-secret
    namespace: default
type: IngressTLS
```

#### **NOTA**

En la información anterior, las **tls.crt** y **tls.key** solo son ejemplos. Reemplácelos con los archivos reales. Los valores de **tls.crt** y **tls.key** están codificados en Base64.

**Paso 3** Cree un secreto.

**kubectl create -f ingress-test-secret.yaml**

Si se muestra información similar a la siguiente, se está creando el secreto:

```
secret/ingress-test-secret created
```

Vea el secreto creado.

**kubectl get secrets**

Si se muestra la información similar a la siguiente, se ha creado el secreto:

NAME	TYPE	DATA	AGE
ingress-test-secret	IngressTLS	2	13s

**Paso 4** Cree un archivo YAML denominado **ingress-test.yaml**. El nombre del archivo se puede personalizar.

**vi ingress-test.yaml**

**Para clústeres de v1.23 y posteriores:**

```

apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress-test
  namespace: default
spec:
  tls:
  - hosts:
    - foo.bar.com
    secretName: ingress-test-secret # Replace it with your TLS key certificate.
  rules:
  - host: ''
    http:
      paths:
      - path: /
        backend:
          service:
            name: <your_service_name> # Replace it with the name of your
target Service.
          port:
            number: <your_service_port> # Replace 8080 with the port
number of your target Service.
        property:
          ingress.beta.kubernetes.io/url-match-mode: STARTS_WITH
        pathType: ImplementationSpecific
    ingressClassName: nginx
    
```

**Para clústeres de v1.21 y anteriores:**

```

apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: ingress-test
  annotations:
    kubernetes.io/ingress.class: nginx
spec:
  tls:
  - hosts:
    - foo.bar.com
    secretName: ingress-test-secret # Replace it with your TLS key certificate.
  rules:
  - host: ''
    http:
      paths:
      - path: '/'
        backend:
          serviceName: <your_service_name> # Replace it with the name of your
target Service.
          servicePort: <your_service_port> # Replace 8080 with the port number
of your target Service.
    ingressClassName: nginx
    
```

**Paso 5** Cree una entrada.

**kubectl create -f ingress-test.yaml**

Si se muestra la información similar a la siguiente, se ha creado la entrada.

```
ingress/ingress-test created
```

Vea la entrada creada.

**kubectl get ingress**

Si se muestra la información similar a la siguiente, se ha creado la entrada y se puede acceder a la carga de trabajo.

NAME	HOSTS	ADDRESS	PORTS	AGE
ingress-test	*	121.**.**.**	80	10s

**Paso 6** Introduzca **https://121.\*\*.\*\*.\*\*:443** en el cuadro de dirección del navegador para acceder a la carga de trabajo (por ejemplo, [Carga de trabajo de Nginx](#)).

**121.\*\*.\*\*.\*\*:** indica la dirección IP del balanceador de carga unificado.

----Fin

### 7.4.3.4 Configuración de reglas de reescritura de URL para ingresos de Nginx

En algunos escenarios de aplicación, el URL de acceso proporcionada por el servicio de backend es diferente de la ruta especificada en la regla de ingreso. La entrada reenvía directamente la ruta de acceso a la misma ruta de backend. Si no se configura la reescritura de URL, se devuelve 404 para todas las solicitudes de acceso. Por ejemplo, si la ruta de acceso en la regla de ingreso se establece en **/app/demo** y la ruta de acceso proporcionada por el servicio de back-end es **/demo**, las solicitudes de acceso se reenvían directamente a la ruta **/app/demo** del servicio de back-end, que no coincide con la ruta de acceso (**/demo** real) proporcionada por el servicio de backend. Como resultado, se devuelve 404.

En este caso, puede utilizar el método Rewrite para implementar la reescritura de URL. Es decir, puede utilizar la anotación **nginx.ingress.kubernetes.io/rewrite-target** para implementar reglas de reescritura para diferentes rutas.

## Configuración de reglas de reescritura

### Para clústeres de v1.23 y posteriores:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress-test
  namespace: default
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /$2
spec:
  rules:
  - host: 'rewrite.bar.com'
    http:
      paths:
      - path: '/something(/|$)(.*)'
        backend:
          service:
            name: <your_service_name> # Replace it with the name of your
target Service.
            port:
              number: <your_service_port> # Replace 8080 with the port
number of your target Service.
          property:
            ingress.beta.kubernetes.io/url-match-mode: STARTS_WITH
            pathType: ImplementationSpecific
        ingressClassName: nginx
```

### Para clústeres de v1.21 y anteriores:

```
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: ingress-test
  namespace: default
  annotations:
    kubernetes.io/ingress.class: nginx
    nginx.ingress.kubernetes.io/rewrite-target: /$2
spec:
  rules:
  - host: 'rewrite.bar.com'
    http:
```

```
paths:
  - path: '/something(/|$)(.*)'
    backend:
      serviceName: <your_service_name> # Replace it with the name of
your target Service.
      servicePort: <your_service_port> # Replace 8080 with the port
number of your target Service.
```

### 📖 NOTA

Mientras se especifique **rewrite-target** para una entrada, todas las rutas bajo el mismo host en todas las definiciones de entrada distinguen entre mayúsculas y minúsculas, incluidas las entradas que no tienen **rewrite-target** especificado.

En el ejemplo anterior, el marcador de posición \$2 indica que todos los caracteres coincidentes con el segundo paréntesis (.) se rellenan en la anotación **nginx.ingress.kubernetes.io/rewrite-target**.

Por ejemplo, la definición de ingreso anterior dará como resultado las siguientes reescrituras:

- **rewrite.bar.com/something** rewrites to **rewrite.bar.com/**.
- **rewrite.bar.com/something/** rewrites to **rewrite.bar.com/**.
- **rewrite.bar.com/something/new** rewrites to **rewrite.bar.com/new**.

En el contenedor **nginx-ingress-controlador**, puede ver todas las configuraciones de ingreso en el archivo **nginx.conf** en el directorio **/etc/nginx**. La regla de reescritura del ejemplo anterior genera un comando Rewrite y lo escribe en el campo **location** del archivo **nginx.conf**.

```
## start server rewrite.bar.com
server {
    server_name rewrite.bar.com ;
    ...
    location ~* "^/something(/|$)(.*)" {
        set $namespace "default";
        set $ingress_name "ingress-test";
        set $service_name "<your_service_name>";
        set $service_port "80";
        ...
        rewrite "(?i)/something(/|$)(.*)" /$2 break;
        ...
    }
}
## end server rewrite.bar.com
```

La sintaxis básica del comando Rewrite es la siguiente:

```
rewrite regex replacement [flag];
```

- **regex**: expresión regular para URI coincidentes. En el ejemplo anterior, **(?i)/something(/|\$)(.)\*** es la expresión regular para URI coincidentes, donde **(?i)** indica que no distingue entre mayúsculas y minúsculas.
- **replacement**: contenido para reescribir. En el ejemplo anterior, el **/\$2** indica que la ruta se reescribe en todos los caracteres coincidentes con el segundo paréntesis **(.)\***.
- **flag**: reescritura de formato.
  - **last**: continúa haciendo coincidir la siguiente regla después de que se haga coincidir la regla actual.
  - **break**: deja de coincidir una vez coincidente la regla actual.
  - **redirect**: devuelve un redireccionamiento temporal con el código 302.
  - **permanent**: devuelve un redireccionamiento permanente con el código 301.

## Configuración avanzada de reescritura

Algunos requisitos complejos y avanzados de Rewrite se pueden implementar modificando el archivo de configuración de Nginx **nginx.conf**. Sin embargo, la función de anotación **nginx.ingress.kubernetes.io/rewrite-target** se puede personalizar para cumplir con requisitos de Rewrite más complejos.

- **nginx.ingress.kubernetes.io/server-snippet**: Agregue una configuración personalizada al campo **server** en el archivo **nginx.conf**.
- **nginx.ingress.kubernetes.io/configuration-snippet**: Agregue una configuración personalizada al campo **location** en el archivo **nginx.conf**.

Puede utilizar las dos anotaciones anteriores para insertar un comando de Rewrite en el campo **server** o **location** del archivo **nginx.conf** para reescribir el URL. A continuación se presenta un ejemplo:

```
annotations:
  kubernetes.io/ingress.class: "nginx"
  nginx.ingress.kubernetes.io/configuration-snippet: |
    rewrite ^/stylesheets/(.*)$ /something/stylesheets/$1 redirect; # Add
the /something prefix.
    rewrite ^/images/(.*)$ /something/images/$1 redirect; # Add the /
something prefix.
```

En las dos reglas anteriores, la ruta **/something** se agrega al URL de acceso.

- Cuando un usuario accede a **rewrite.bar.com/stylesheets/new.css** vuelve a escribir en **rewrite.bar.com/something/stylesheets/new.css**.
- Cuando un usuario accede a **rewrite.bar.com/images/new.jpg** vuelve a escribir en **rewrite.bar.com/something/images/new.jpg**.

## Redireccionando HTTP a HTTPS

De forma predeterminada, si una entrada utiliza TLS, las solicitudes se redirigirán (código de estado 308) a HTTPS cuando se utilice HTTP para el acceso. También puede utilizar la siguiente anotación para redirigir a la fuerza las solicitudes a HTTPS.

### Para clústeres de v1.23 y posteriores:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress-test
  namespace: default
  annotations:
    nginx.ingress.kubernetes.io/ssl-redirect: 'true'
spec:
  rules:
    - host: ''
      http:
        paths:
          - path: /
            backend:
              service:
                name: <your_service_name> # Replace it with the name of your
target Service.
            port:
              number: <your_service_port> # Replace 8080 with the port
number of your target Service.
          property:
            ingress.beta.kubernetes.io/url-match-mode: STARTS_WITH
          pathType: ImplementationSpecific
        ingressClassName: nginx
```

**Para clústeres de v1.21 y anteriores:**

```
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: ingress-test
  namespace: default
  annotations:
    kubernetes.io/ingress.class: nginx
    nginx.ingress.kubernetes.io/ssl-redirect: 'true'
spec:
  rules:
  - host: ''
    http:
      paths:
      - path: /
        backend:
          serviceName: <your_service_name> # Replace it with the name of
your target Service.
          servicePort: <your_service_port> # Replace 8080 with the port
number of your target Service.
```

### 7.4.3.5 Interconexión de ingresos de Nginx con servicios de backend HTTPS

Ingreso puede funcionar como un proxy para servicios de back-end usando diferentes protocolos. De forma predeterminada, el canal proxy backend de un ingreso es un canal HTTP. Para crear un canal HTTPS, agregue la siguiente configuración al campo **annotations**:

```
nginx.ingress.kubernetes.io/backend-protocol: "HTTPS"
```

Un ejemplo de configuración de ingreso:

**Para clústeres de v1.23 y posteriores:**

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress-test
  namespace: default
  annotations:
    nginx.ingress.kubernetes.io/backend-protocol: "HTTPS"
spec:
  tls:
  - secretName: ingress-test-secret # Replace it with your TLS key certificate.
  rules:
  - host: ''
    http:
      paths:
      - path: '/'
        backend:
          service:
            name: <your_service_name> # Replace it with the name of your
target Service.
            port:
              number: <your_service_port> # Replace 8080 with the port
number of your target Service.
          property:
            ingress.beta.kubernetes.io/url-match-mode: STARTS_WITH
            pathType: ImplementationSpecific
        ingressClassName: nginx
```

**Para clústeres de v1.21 y anteriores:**

```
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: ingress-test
  namespace: default
  annotations:
    kubernetes.io/ingress.class: nginx
```

```
    nginx.ingress.kubernetes.io/backend-protocol: "HTTPS"
spec:
  tls:
    - secretName: ingress-test-secret # Replace it with your TLS key certificate.
  rules:
    - host: ''
      http:
        paths:
          - path: '/'
            backend:
              serviceName: <your_service_name> # Replace it with the name of
your target Service.
              servicePort: <your_service_port> # Replace 8080 with the port
number of your target Service.
```

### 7.4.3.6 Ingresos de Nginx usando hashing consistente para el balanceo de carga

El Nginx nativo soporta múltiples reglas de balanceo de carga, incluidos round robin ponderado y hash IP. Un ingreso de Nginx soporta el balanceo de carga mediante el uso de hash consistente basado en las capacidades nativas de Nginx.

De forma predeterminada, el método hash IP soportado por Nginx utiliza el espacio hash lineal. El servidor backend se selecciona en función del valor hash de la dirección IP. Sin embargo, cuando se utiliza este método para agregar o eliminar un nodo, todas las direcciones IP deben ser hash de nuevo y luego enrutadas de nuevo. Como resultado, se pierde un gran número de sesiones o la caché se vuelve inválida. Por lo tanto, se introduce un hash consistente en la entrada de Nginx para resolver este problema.

El hash consistente es un algoritmo hash especial, que construye un espacio hash de anillo para reemplazar el espacio hash lineal común. Cuando se agrega o elimina un nodo, solo se migra la ruta de destino en el sentido de las agujas del reloj, y no es necesario cambiar otras rutas. De esta manera, el reencaminamiento puede reducirse tanto como sea posible, resolviendo el problema de balanceo de carga causado por la adición y eliminación de nodos dinámicos.

Si se configura una regla de hash consistente, el servidor recién agregado compartirá la carga de todos los demás servidores. Del mismo modo, cuando se elimina un servidor, todos los demás servidores pueden compartir la carga del servidor eliminado. Esto equilibra la carga entre nodos en el clúster y evita el efecto de avalancha causado por la avería de un nodo.

## Configuración de una regla de hash coherente

Una entrada Nginx puede usar la anotación `nginx.ingress.kubernetes.io/upstream-hash-by` para configurar reglas de hash consistentes. A continuación se presenta un ejemplo:

### Para clústeres de v1.23 y posteriores:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress-test
  namespace: default
  annotations:
    nginx.ingress.kubernetes.io/upstream-hash-by: "$request_uri" # Perform
hashing based on the request URI.
spec:
  rules:
    - host: ''
      http:
        paths:
          - path: '/'
            backend:
              service:
```

```
name: <your_service_name> # Replace it with the name of your
target Service.
port:
  number: <your_service_port> # Replace 8080 with the port
number of your target Service.
property:
  ingress.beta.kubernetes.io/url-match-mode: STARTS_WITH
pathType: ImplementationSpecific
ingressClassName: nginx
```

#### Para clústeres de v1.21 y anteriores:

```
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: ingress-test
  namespace: default
  annotations:
    kubernetes.io/ingress.class: nginx
    nginx.ingress.kubernetes.io/upstream-hash-by: "$request_uri" # Perform
hashing based on the request URI.
spec:
  rules:
    - host: ''
      http:
        paths:
          - path: '/'
            backend:
              serviceName: <your_service_name> # Replace it with the name of
your target Service.
              servicePort: <your_service_port> # Replace 8080 with the port
number of your target Service.
```

El valor de **nginx.ingress.kubernetes.io/upstream-hash-by** puede ser una variable de nginx, un valor de texto o cualquier combinación:

- **nginx.ingress.kubernetes.io/upstream-hash-by: "\$request\_uri"** indica que las solicitudes son hash basadas en el URI de solicitud.
- **nginx.ingress.kubernetes.io/upstream-hash-by: "\$request\_uri\$host"** indica que las solicitudes son hash basadas en el URI de solicitud y el nombre de dominio.
- **nginx.ingress.kubernetes.io/upstream-hash-by: "\${request\_uri}-text-value"** indica que las solicitudes son hash basadas en el URI de solicitud y el valor de texto.

## Documentación

### [Custom NGINX upstream hashing](#)

#### 7.4.3.7 Configuración de ingresos de Nginx con anotaciones

El complemento nginx-ingress en CCE utiliza el gráfico y la imagen de la comunidad. Si los parámetros de complemento predeterminados no pueden satisfacer sus demandas, puede agregar anotaciones para definir lo que necesita, como el backend predeterminado, el tiempo de espera y el tamaño del cuerpo de una solicitud.

Esta sección describe las anotaciones comunes utilizadas para crear una entrada del tipo de Nginx.

#### NOTA

El valor clave de una anotación solo puede ser una cadena. Otros tipos (como valores Booleano o valores numéricos) deben estar entre comillas (""), por ejemplo, "true", "false" y "100".

## Tipo de ingreso

**Tabla 7-33** Anotaciones de tipo de ingreso

Parámetro	Tipo	Descripción	Versión de clúster admitida
kubernetes.io/ingress.class	String	<ul style="list-style-type: none"> <li>● <b>nginx</b>: Se utiliza la entrada de Nginx.</li> <li>● <b>cce</b>: Se utiliza la entrada de ELB autodesarrollada.</li> </ul> Este parámetro es obligatorio cuando se crea una entrada llamando a la API. Para los clústeres de v1.23 o posterior, utilice el parámetro <b>ingressClassName</b> . Para obtener más información, véase <a href="#">Uso de kubectl para crear una entrada de Nginx</a> .	Solo clústeres de v1.21 o anteriores

Para obtener más información acerca de cómo utilizar las anotaciones anteriores, vea [Uso de kubectl para crear una entrada de Nginx](#).

## Configuración de una regla de reescritura de URL

**Tabla 7-34** Anotaciones de regla de reescritura de URL

Parámetro	Tipo	Descripción
nginx.ingress.kubernetes.io/rewrite-target	String	URI de destino donde el tráfico debe ser redirigido.
nginx.ingress.kubernetes.io/ssl-redirect	Bool	Indica si el acceso solo está disponible con SSL. El valor predeterminado es <b>true</b> cuando la entrada contiene un certificado.
nginx.ingress.kubernetes.io/force-ssl-redirect	Bool	Indica si se debe redirigir a la fuerza una solicitud a HTTPS incluso si TLS no está habilitado para la entrada. Cuando se utiliza HTTP para el acceso, la solicitud se redirige a la fuerza (código de estado 308) a HTTPS.

Para obtener más información sobre los escenarios de la aplicación, consulte [Configuración de reglas de reescritura de URL para ingresos de Nginx](#).

## Interconexión con servicios de backend HTTPS

**Tabla 7-35** Anotaciones de interconexión con servicios backend HTTPS

Parámetro	Tipo	Descripción
nginx.ingress.kubernetes.io/backend-protocol	String	Si este parámetro se establece en <b>HTTPS</b> , se utiliza HTTPS para reenviar solicitudes al contenedor de servicio de back-end.

Para obtener más información sobre los escenarios de la aplicación, consulte [Interconexión de ingresos de Nginx con servicios de backend HTTPS](#).

## Creación de una regla de hash coherente para balanceo de carga

**Tabla 7-36** Anotación de hash consistente para el balanceo de carga

Parámetro	Tipo	Descripción
nginx.ingress.kubernetes.io/upstream-hash-by	String	<p>Habilite un hash consistente para equilibrar la carga de los servidores backend. El valor del parámetro puede ser un parámetro de nginx, un valor de texto o cualquier combinación. Por ejemplo:</p> <ul style="list-style-type: none"> <li>● <b>nginx.ingress.kubernetes.io/upstream-hash-by: "\$request_uri"</b> indica que las solicitudes son hash basadas en el URI de solicitud.</li> <li>● <b>nginx.ingress.kubernetes.io/upstream-hash-by: "\$request_uri\$host"</b> indica que las solicitudes son hash basadas en el URI de solicitud y el nombre de dominio.</li> <li>● <b>nginx.ingress.kubernetes.io/upstream-hash-by: "\${request_uri}-text-value"</b> indica que las solicitudes son hash basadas en el URI de solicitud y el valor de texto.</li> </ul>

Para obtener más información sobre los escenarios de la aplicación, consulte [Ingresos de Nginx usando hashing consistente para el balanceo de carga](#).

## Intervalo de tiempo de espera personalizado

**Tabla 7-37** Anotaciones de intervalo de tiempo de espera personalizadas

Parámetro	Tipo	Descripción
nginx.ingress.kubernetes.io/proxy-connect-timeout	String	Intervalo de tiempo de espera de conexión personalizado. No es necesario ajustar la unidad al establecer el intervalo de tiempo de espera. La unidad predeterminada es la segunda.  Por ejemplo: nginx.ingress.kubernetes.io/proxy-connect-timeout: '120'

## Personalización del tamaño del cuerpo

**Tabla 7-38** Anotaciones de personalización del tamaño del cuerpo

Parámetro	Tipo	Descripción
nginx.ingress.kubernetes.io/proxy-body-size	String	Cuando el tamaño del cuerpo en una solicitud excede el límite superior, el error 413 se devuelve al cliente. Puede utilizar este parámetro para ajustar el límite superior del tamaño del cuerpo.  Por ejemplo: nginx.ingress.kubernetes.io/proxy-body-size: 8m

## Documentación

Para obtener más información sobre los parámetros de anotación admitidos por las entradas de Nginx, consulte [Anotaciones](#).

## 7.5 DNS

### 7.5.1 Descripción general

#### Introducción a CoreDNS

Cuando se crea un clúster, el [complemento de coredns](#) se instala para resolver nombres de dominio en el clúster.

Puede ver el pod del complemento de coredns en el espacio de nombres del kube-system.

```
$ kubectl get po --namespace=kube-system
NAME                                READY   STATUS    RESTARTS   AGE
coredns-7689f8bdf-295rk            1/1     Running   0           9m11s
coredns-7689f8bdf-h7n68            1/1     Running   0           11m
```

Después de instalar coredns, se convierte en un DNS. Después de crear el Service, coredns registra el nombre del Service y la dirección IP. De esta manera, el pod puede obtener la dirección IP del Service consultando el nombre del Service desde coredns.

**nginx.<namespace>.svc.cluster.local** se utiliza para acceder al Service. **nginx** es el nombre del Service, **<namespace>** es el espacio de nombres y **svc.cluster.local** es el sufijo del nombre de dominio. En uso real, puede omitir **<namespace>.svc.cluster.local** en el mismo espacio de nombres y usar el ServiceName.

Una ventaja de usar ServiceName es que puede escribir ServiceName en el programa al desarrollar la aplicación. De esta manera, no es necesario conocer la dirección IP de un Service específico.

Después de instalar el complemento de coredns, también hay un Service en el espacio de nombres de kube-system, como se muestra a continuación.

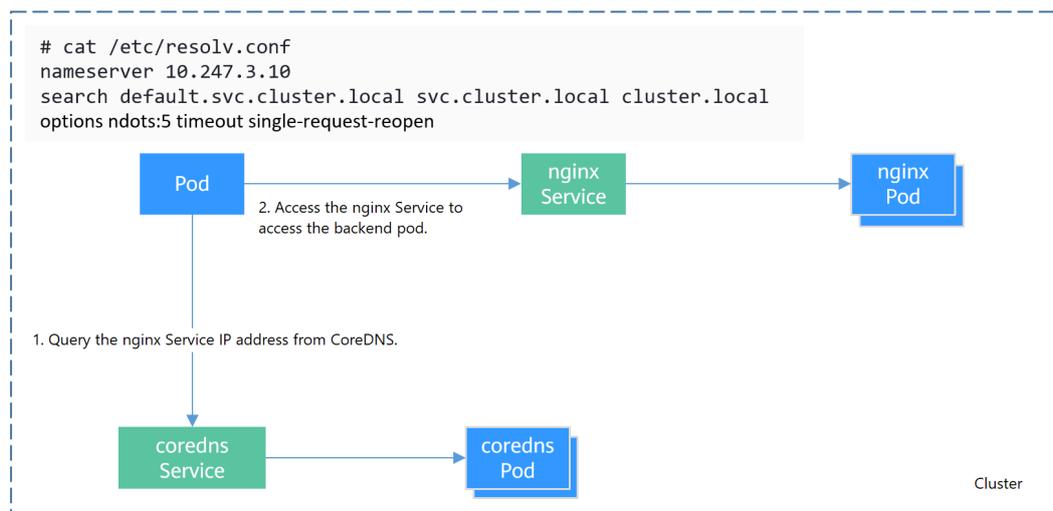
```
$ kubectl get svc -n kube-system
NAME                                TYPE                CLUSTER-IP      EXTERNAL-IP
PORT(S)                              AGE                CLUSTER-IP      EXTERNAL-IP
coredns                               ClusterIP           10.247.3.10     <none>
53/UDP, 53/TCP, 8080/TCP             13d
```

De forma predeterminada, después de crear otros pods, la dirección del Service de coredns se escribe como la dirección del servidor de resolución de nombres de dominio en el archivo **/etc/resolv.conf** del pod. Cree un pod y vea el archivo **/etc/resolv.conf** de la siguiente manera:

```
$ kubectl exec test01-6cbbf97b78-krj6h -it -- /bin/sh
/ # cat /etc/resolv.conf
nameserver 10.247.3.10
search default.svc.cluster.local svc.cluster.local cluster.local
options ndots:5 timeout single-request-reopen
```

Cuando un usuario accede al *Service name:Port* del pod Nginx, la dirección IP del Service Nginx se resuelve desde CoreDNS y, a continuación, se accede a la dirección IP del Service Nginx. De esta manera, el usuario puede acceder al pod backend de Nginx.

**Figura 7-32** Ejemplo de resolución de nombres de dominio en un clúster



## Operaciones relacionadas

También puede configurar DNS en una carga de trabajo. Para obtener más información, véase [Configuración de DNS](#).

También puede usar `coredns` para implementar la resolución de nombres de dominio definida por el usuario. Para obtener más información, véase [Uso de CoreDNS para la resolución personalizada de nombres de dominio](#).

También puede utilizar `DNSCache` para mejorar el rendimiento de la resolución de DNS. Para obtener más información, véase [Uso de DNSCache de NodeLocal para mejorar el rendimiento de DNS](#).

## 7.5.2 Configuración de DNS

Cada clúster de Kubernetes tiene un complemento de DNS integrado (Kube-DNS o CoreDNS) para proporcionar resolución de nombres de dominio para las cargas de trabajo del clúster. Cuando se maneja una alta concurrencia de consultas de DNS, Kube-DNS/CoreDNS puede encontrar un cuello de botella de rendimiento, es decir, puede fallar ocasionalmente para satisfacer las consultas de DNS. Hay casos en los que las cargas de trabajo de Kubernetes inician consultas de DNS innecesarias. Esto hace que el DNS se sobrecargue si hay muchas consultas de DNS simultáneas. Al ajustar la configuración de DNS para las cargas de trabajo, se reducirán en cierta medida los riesgos de fallas en las consultas de DNS.

Para obtener más información acerca de DNS, consulte [coredns \(complemento de recursos del sistema, obligatorio\)](#).

## Conceptos de configuración de DNS

Ejecute el comando `cat /etc/resolv.conf` en un nodo de Linux o contenedor para ver el archivo de configuración del solucionador de DNS. A continuación se muestra un ejemplo de configuración de resolución de DNS de un contenedor en un clúster de Kubernetes:

```
nameserver 10.247.x.x
search default.svc.cluster.local svc.cluster.local cluster.local
options ndots:5
```

### Opciones de configuración

- **nameserver**: una lista de direcciones IP de un servidor de nombres que el solucionador consultará. Si este parámetro se establece en `10.247.x.x`, el solucionador consultará el `kube-dns/CoreDNS`. Si este parámetro se establece en otra dirección IP, el solucionador consultará un servidor DNS en la nube o local.
- **search**: una lista de búsqueda para la búsqueda de nombre de host. Cuando un nombre de dominio no se puede resolver, las consultas de DNS se intentarán combinando el nombre de dominio con cada dominio en la lista de búsqueda a su vez hasta que se encuentre una coincidencia o se intenten todos los dominios en la lista de búsqueda. En el caso de los clústeres de CCE, la lista de búsqueda se limita actualmente a tres dominios por contenedor. Cuando se resuelve un nombre de dominio inexistente, se iniciarán ocho consultas de DNS porque cada nombre de dominio (incluidos los de la lista de búsqueda) se consultará dos veces, una para IPv4 y la otra para IPv6.
- **options**: opciones que permiten modificar ciertas variables internas de resolución. Las opciones comunes incluyen el tiempo de espera y los `ndots`.

El valor `ndots:5` significa que si un nombre de dominio tiene menos de 5 puntos (`.`), las consultas de DNS se intentarán combinando el nombre de dominio con cada dominio en la lista de búsqueda a su vez. Si no se encuentra ninguna coincidencia después de probar

todos los dominios de la lista de búsqueda, el nombre de dominio se utiliza para la consulta de DNS. Si el nombre de dominio tiene 5 o más de 5 puntos, se intentará primero para la consulta de DNS. En caso de que el nombre de dominio no se pueda resolver, las consultas de DNS se intentarán combinando el nombre de dominio con cada dominio en la lista de búsqueda a su vez.

Por ejemplo, el nombre de dominio **www.\*\*\*.com** tiene solo dos puntos (más pequeño que el valor de **ndots**) y, por lo tanto, la secuencia de consultas de DNS es como las siguientes: **www.\*\*\*.default.svc.cluster.local**, **www.\*\*\*.com.svc.cluster.local**, **www.\*\*\*.com.cluster.local** y **www.\*\*\*.com**. Esto significa que se iniciarán al menos siete consultas de DNS antes de que el nombre de dominio se resuelva en una dirección IP. Está claro que cuando se iniciarán muchas consultas de DNS innecesarias para acceder a un nombre de dominio externo. Hay margen para mejorar la configuración de DNS de la carga de trabajo.

### NOTA

Para obtener más información acerca de las opciones de configuración en el archivo de configuración de resolución utilizado por los sistemas operativos Linux, visite <http://man7.org/linux/man-pages/man5/resolv.conf.5.html>.

## Configuración de DNS mediante la carga de trabajo de YAML

Al crear una carga de trabajo con un archivo YAML, puede configurar la configuración de DNS en YAML. El siguiente es un ejemplo para una aplicación de Nginx:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: container-1
          image: nginx:latest
          imagePullPolicy: IfNotPresent
      imagePullSecrets:
        - name: default-secret
      dnsPolicy: None
      dnsConfig:
        options:
          - name: ndots
            value: '5'
          - name: timeout
            value: '3'
        nameservers:
          - 10.2.3.4
        searches:
          - my.dns.search.suffix
```

### **dnsPolicy**

El campo **dnsPolicy** se utiliza para configurar una política de DNS para una aplicación. El valor predeterminado es **ClusterFirst**. Los parámetros de DNS de **dnsConfig** se fusionarán

con el archivo de DNS generado según **dnsPolicy**. Las reglas de fusión se explican más adelante en **Tabla 7-40**. Actualmente, **dnsPolicy** admite los cuatro valores siguientes:

**Tabla 7-39** dnsPolicy

Parámetro	Descripción
ClusterFirst (default value)	El CoreDNS del clúster de CCE, que se conecta en cascada con el DNS en la nube de forma predeterminada, se utiliza para las cargas de trabajo. Los contenedores pueden resolver tanto los nombres de dominio internos de clúster registrados por un Service como los nombres de dominio externos expuestos a las redes públicas. La lista de búsqueda (opción <b>search</b> ) y <b>ndots: 5</b> están presentes en el archivo de configuración de DNS. Por lo tanto, al acceder a un nombre de dominio externo y a un nombre de dominio interno de clúster largo (por ejemplo, <code>kubernetes.default.svc.cluster.local</code> ), la lista de búsqueda generalmente se recorrerá primero, resultando en al menos seis consultas de DNS no válidas. El problema de las consultas de DNS no válidas desaparece solo cuando se accede a un nombre de dominio interno de clúster corto (por ejemplo, <code>kubernetes</code> ).
ClusterFirstWithHostNet	De forma predeterminada, el archivo de configuración de DNS al que apunta el indicador <b>--resolv-conf</b> está configurado para las cargas de trabajo que se ejecutan con <b>hostNetwork=true</b> es decir, se utiliza un DNS en la nube para los clústeres de CCE. Si las cargas de trabajo necesitan usar Kube-DNS/CoreDNS del clúster, establezca <b>dnsPolicy</b> en <b>ClusterFirstWithHostNet</b> y el archivo de configuración de DNS de contenedor es el mismo que ClusterFirst en el que aún existen las consultas de DNS no válidas. <pre> ... spec:   containers:   - image: nginx:latest     imagePullPolicy: IfNotPresent     name: container-1     restartPolicy: Always     hostNetwork: true     dnsPolicy: ClusterFirstWithHostNet </pre>
Default	El archivo de configuración de DNS del contenedor es el archivo de configuración de DNS al que apunta el indicador <b>--resolv-conf</b> del kubelet. En este caso, se utiliza un DNS en la nube para los clústeres de CCE. Tanto los campos <b>search</b> como <b>options</b> se dejan sin especificar. Esta configuración solo puede resolver los nombres de dominio externos registrados con Internet, y no los nombres de dominio internos de clúster. Esta configuración está libre de la emisión de consultas de DNS no válidas.
None	Si <b>dnsPolicy</b> está establecido en <b>None</b> debe especificarse el campo <b>dnsConfig</b> porque se supone que todas las configuraciones de DNS se proporcionan mediante el campo <b>dnsConfig</b> .

 **NOTA**

Si no se especifica el campo **dnsPolicy**, el valor predeterminado es **ClusterFirst** en lugar de **Default**.

## dnsConfig

El campo **dnsConfig** se utiliza para configurar los parámetros de DNS para cargas de trabajo. Los parámetros configurados se fusionan con el archivo de configuración de DNS generado según **dnsPolicy**. Si **dnsPolicy** se establece en **None**, el campo **dnsConfig** especifica el archivo de configuración de DNS de la carga de trabajo. Si **dnsPolicy** no se establece en **None**, los parámetros de DNS configurados en el **dnsConfig** se agregan al archivo de configuración de DNS generado de acuerdo con la norma **dnsPolicy**.

Tabla 7-40 dnsConfig

Parámetro	Descripción
options	Una lista opcional de objetos donde cada objeto puede tener una propiedad name (obligatoria) y una propiedad value (opcional). El contenido de esta propiedad se combinará con las opciones generadas a partir de la política de DNS especificada de <b>dnsPolicy</b> . Se eliminan las entradas duplicadas.
nameservers	Una lista de direcciones IP que se utilizarán como servidores de DNS. Si <b>dnsPolicy</b> de la carga de trabajo se establece en <b>None</b> , la lista debe contener al menos una dirección IP; de lo contrario, esta propiedad es opcional. Los servidores listados se combinarán con los servidores de nombres generados a partir de la política de DNS especificada de <b>dnsPolicy</b> con las direcciones duplicadas quitadas.
searches	Una lista de dominios de búsqueda de DNS para la búsqueda de nombre de host en el Pod. Esta propiedad es opcional. Cuando se especifica, la lista proporcionada se fusionará con los nombres de dominio de búsqueda generados a partir de la política de DNS elegida de <b>dnsPolicy</b> . Se quitan los nombres de dominio duplicados. Kubernetes permite un máximo de 6 dominios de búsqueda.

## Configuración de DNS para una carga de trabajo mediante la consola

Kubernetes proporciona opciones de configuración relacionadas con DNS para las aplicaciones. El uso de la configuración de DNS de la aplicación puede reducir eficazmente las consultas DNS innecesarias en ciertos escenarios y mejorar la concurrencia del servicio. En el siguiente procedimiento se utiliza una aplicación de Nginx como ejemplo para describir cómo agregar configuraciones de DNS para una carga de trabajo en la consola.

- Paso 1** Inicie sesión en la consola de CCE, acceda a la consola del clúster, seleccione **Workloads** en el panel de navegación y haga clic en **Create Workload** en la esquina superior derecha.
- Paso 2** Configure la información básica sobre la carga de trabajo. Para obtener más información, véase [Creación de una Deployment](#).
- Paso 3** En el área **Advanced Settings**, haga clic en la ficha **DNS** y establezca los siguientes parámetros según sea necesario:
  - **DNS Policy**: Las políticas de DNS proporcionadas en la consola corresponden al campo **dnsPolicy** del archivo YAML. Para obtener más información, véase [Tabla 7-39](#).

- **Supplement defaults:** corresponde a **dnsPolicy=ClusterFirst**. Los contenedores pueden resolver tanto los nombres de dominio internos de clúster registrados por un Service como los nombres de dominio externos expuestos a las redes públicas.
- **Replace defaults:** corresponde a **dnsPolicy=None**. Debe configurar **IP Address** y **Search Domain**. Los contenedores solo utilizan la dirección IP definida por el usuario y las configuraciones de dominio de búsqueda para la resolución de nombres de dominio.
- **Inherit defaults:** corresponde a **dnsPolicy=Default**. Los contenedores utilizan la configuración de resolución de nombres de dominio desde el nodo en el que se ejecutan los pods y no pueden resolver los nombres de dominio internos del clúster.
- **Optional Objects:** los parámetros de opciones del **campo de dnsConfig**. Cada objeto puede tener una propiedad del nombre (requerida) y una propiedad del valor (opcional). Después de establecer las propiedades, haga clic en **confirm to add**.
  - **timeout:** Intervalo de tiempo de espera, en segundos.
  - **ndots:** Número de puntos (.) que deben estar presentes en un nombre de dominio. Si un nombre de dominio tiene puntos menores que este valor, el sistema operativo buscará el nombre en el dominio de búsqueda. Si no, el nombre es un nombre de dominio completo (FQDN) y se probará primero como un nombre absoluto.
- **IP Address: nameservers** en **dnsConfig**. Puede configurar el servidor de nombres de dominio para el nombre de dominio personalizado. El valor es una o un grupo de direcciones IP de DNS.
- **Search Domain: searches** en **dnsConfig**. Una lista de dominios de búsqueda de DNS para la búsqueda de nombre de host en el pod. Esta propiedad es opcional. Cuando se especifica, la lista proporcionada se fusionará con los nombres de dominio de búsqueda generados a partir de la política de DNS elegida de **dnsPolicy**. Se quitan los nombres de dominio duplicados.

#### Paso 4 Hacer clic en **Create Workload**.

----Fin

## Ejemplos de configuración

En el ejemplo siguiente se describe cómo configurar DNS para las cargas de trabajo.

- **Caso del uso 1: Uso de Kube-DNS/CoreDNS construido en los clústeres de Kubernetes**

### Escenario

Kubernetes en clúster Kube-DNS/CoreDNS es aplicable para resolver solo nombres de dominio internos de clúster o nombres de dominio internos de clúster + nombres de dominio externos. Este es el DNS predeterminado para las cargas de trabajo.

**Por ejemplo:**

```
apiVersion: v1
kind: Pod
metadata:
  namespace: default
  name: dns-example
spec:
  containers:
  - name: test
    image: nginx:alpine
  dnsPolicy: ClusterFirst
```

Archivo de configuración de DNS del contenedor:

```
nameserver 10.247.3.10
search default.svc.cluster.local svc.cluster.local cluster.local
options ndots:5
```

- **Caso de uso 2: Uso de un DNS en la nube**

**Escenario**

Un DNS no puede resolver nombres de dominio internos de clúster y, por lo tanto, es aplicable al escenario en el que las cargas de trabajo acceden solo a los nombres de dominio externos registrados con Internet.

**Por ejemplo:**

```
apiVersion: v1
kind: Pod
metadata:
  namespace: default
  name: dns-example
spec:
  containers:
  - name: test
    image: nginx:alpine
  dnsPolicy: Default//The DNS configuration file that the kubelet's --resolv-conf flag points to is used. In this case, a DNS is used for CCE clusters.
```

Archivo de configuración de DNS del contenedor:

```
nameserver 100.125.x.x
```

- **Caso de uso 3: Uso de Kube-DNS/CoreDNS para cargas de trabajo que se ejecutan con hostNetwork**

**Escenario**

De forma predeterminada, se utiliza un DNS para las cargas de trabajo que se ejecutan con hostNetwork. Si las cargas de trabajo necesitan usar Kube-DNS/CoreDNS, establezca **dnsPolicy** en **ClusterFirstWithHostNet**.

**Por ejemplo:**

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  hostNetwork: true
  dnsPolicy: ClusterFirstWithHostNet
  containers:
  - name: nginx
    image: nginx:alpine
    ports:
    - containerPort: 80
```

Archivo de configuración de DNS del contenedor:

```
nameserver 10.247.3.10
search default.svc.cluster.local svc.cluster.local cluster.local
options ndots:5
```

- **Caso del uso 4: Personalización de la configuración de DNS de la aplicación**

### Escenario

Puede personalizar de forma flexible el archivo de configuración de DNS para las aplicaciones. El uso de **dnsPolicy** y **dnsConfig** juntos puede abordar casi todos los escenarios, incluidos los escenarios en los que se usará un DNS local, varios DNS se conectarán en cascada y las opciones de configuración de DNS se modificarán.

### Ejemplo 1: Uso de su DNS local

Establezca **dnsPolicy** a **None** para que el archivo de configuración DNS de la aplicación se genere basado en **dnsConfig**.

```
apiVersion: v1
kind: Pod
metadata:
  namespace: default
  name: dns-example
spec:
  containers:
  - name: test
    image: nginx:alpine
    dnsPolicy: "None"
    dnsConfig:
      nameservers:
      - 10.2.3.4 //IP address of your on-premises DNS
      searches:
      - ns1.svc.cluster.local
      - my.dns.search.suffix
      options:
      - name: ndots
        value: "2"
      - name: timeout
        value: "3"
```

Archivo de configuración de DNS del contenedor:

```
nameserver 10.2.3.4
search ns1.svc.cluster.local my.dns.search.suffix
options timeout:3 ndots:2
```

### Ejemplo 2: Modificación de la opción de ndots en el archivo de configuración de DNS para reducir las consultas de DNS no válidas

Establezca **dnsPolicy** en un valor distinto de **None** para que los parámetros de DNS configurados en **dnsConfig** se agreguen al archivo de configuración de DNS generado según **dnsPolicy**.

```
apiVersion: v1
kind: Pod
metadata:
  namespace: default
  name: dns-example
spec:
  containers:
  - name: test
    image: nginx:alpine
    dnsPolicy: "ClusterFirst"
    dnsConfig:
      options:
      - name: ndots
        value: "2" //Changes the ndots:5 option in the DNS configuration file
        generated based on the ClusterFirst policy to ndots:2.
```

Archivo de configuración de DNS del contenedor:

```
nameserver 10.247.3.10
search default.svc.cluster.local svc.cluster.local cluster.local
options ndots:2
```

## 7.5.3 Uso de CoreDNS para la resolución personalizada de nombres de dominio

### Desafíos

Al usar CCE, es posible que deba resolver nombres de dominio internos personalizados en los siguientes escenarios:

- En el código heredado, se configura un nombre de dominio fijo para invocar a otros servicios internos. Si el sistema decide usar Kubernetes Services, la carga de trabajo de refactorización de código podría ser pesada.
- Se crea un servicio fuera del clúster. Los datos del clúster deben enviarse al servicio con un nombre de dominio fijo.

### Solución

Existen varias soluciones basadas en CoreDNS para la resolución personalizada de nombres de dominio:

- **Configuración del dominio Stub para CoreDNS:** Puede agregarlo en la consola, que es fácil de operar.
- **Uso del complemento Hosts de CoreDNS para configurar la resolución de cualquier nombre de dominio:** Puede agregar cualquier conjunto de registros, lo cual es similar a agregar un conjunto de registros en el archivo `/etc/hosts` local.
- **Uso del complemento Rewrite de CoreDNS para apuntar un nombre de dominio a un servicio en el clúster:** Se asigna un apodo al Kubernetes Service. No es necesario conocer de antemano la dirección IP del registro de resolución.
- **Uso del complemento Forward de CoreDNS para configurar el DNS autoconstruido como DNS ascendente:** El DNS autoconstruido puede gestionar un gran número de registros de resolución. No es necesario modificar la configuración de CoreDNS al agregar o eliminar registros.

### Precauciones

La modificación incorrecta en la configuración de CoreDNS puede provocar errores de resolución de nombres de dominio en el clúster. Realice pruebas antes y después de la modificación.

### Configuración del dominio Stub para CoreDNS

Los administradores de clústeres pueden modificar el ConfigMap para CoreDNS Corefile para cambiar el funcionamiento de la detección de servicios.

Suponga que un administrador de clúster tiene un servidor de Consul DNS ubicado en 10.150.0.1 y que todos los nombres de dominio Consul tienen el sufijo `.consul.local`.

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** En el panel de navegación, elija **Add-ons**. En la página mostrada, haga clic en **Edit** en CoreDNS.

**Paso 3** Agregue un dominio stub en el área **Parameters**.

Modifique el parámetro **stub\_domains** en el formato de un par clave-valor. La clave es un nombre de dominio de sufijo de DNS y el valor es una dirección IP de DNS o un grupo de direcciones IP de DNS.

```
{
  "stub_domains": {
    "consul.local": [
      "10.150.0.1"
    ]
  },
  "upstream_nameservers": []
}
```

**Paso 4** Haga clic en **OK**.

---Fin

También puede modificar el ConfigMap de la siguiente manera:

### AVISO

Los valores de los parámetros en rojo en el ejemplo solo se pueden modificar y no se pueden eliminar.

```
$ kubectl edit configmap coredns -n kube-system
apiVersion: v1
data:
  Corefile: |-
    .:5353 {
      bind {$POD_IP}
      cache 30
      errors
      health {$POD_IP}:8080
      kubernetes cluster.local in-addr.arpa ip6.arpa {
        pods insecure
        fallthrough in-addr.arpa ip6.arpa
      }
      loadbalance round_robin
      prometheus {$POD_IP}:9153
      forward . /etc/resolv.conf {
        policy random
      }
      reload
    }

    consul.local:5353 {
      bind {$POD_IP}
      errors
      cache 30
      forward . 10.150.0.1
    }
kind: ConfigMap
metadata:
  creationTimestamp: "2022-05-04T04:42:24Z"
  labels:
    app: coredns
    k8s-app: coredns
    kubernetes.io/cluster-service: "true"
    kubernetes.io/name: CoreDNS
    release: cceaddon-coredns
  name: coredns
  namespace: kube-system
  resourceVersion: "8663493"
  uid: bba87142-9f8d-4056-b8a6-94c3887e9e1d
```

## Modificación del archivo de configuración hosts de CoreDNS

**Paso 1** Utilice kubectl para conectarse al clúster.

**Paso 2** Modifique el archivo de configuración de CoreDNS y agregue el nombre de dominio personalizado al archivo hosts.

Apunte **www.example.com** a **192.168.1.1**. Cuando CoreDNS resuelve **www.example.com**, **192.168.1.1** se devuelve.

### AVISO

Se debe configurar el campo `fallthrough`. **fallthrough** indica que cuando el nombre de dominio a resolver no se encuentra en el archivo `hosts`, la tarea de resolución se transfiere al siguiente complemento de CoreDNS. Si no se especifica **fallthrough**, la tarea finaliza y la resolución del nombre de dominio se detiene. Como resultado, la resolución de nombres de dominio en el clúster falla.

Para obtener más información sobre cómo configurar el archivo `hosts`, visite <https://coredns.io/plugins/hosts/>.

```
$ kubectl edit configmap coredns -n kube-system
apiVersion: v1
data:
  Corefile: |-
    .:5353 {
      bind {$POD_IP}
      cache 30
      errors
      health {$POD_IP}:8080
      kubernetes cluster.local in-addr.arpa ip6.arpa {
        pods insecure
        fallthrough in-addr.arpa ip6.arpa
      }
      hosts {
        192.168.1.1 www.example.com
        fallthrough
      }
      loadbalance round_robin
      prometheus {$POD_IP}:9153
      forward . /etc/resolv.conf
      reload
    }
kind: ConfigMap
metadata:
  creationTimestamp: "2021-08-23T13:27:28Z"
  labels:
    app: coredns
    k8s-app: coredns
    kubernetes.io/cluster-service: "true"
    kubernetes.io/name: CoreDNS
    release: cceaddon-coredns
  name: coredns
  namespace: kube-system
  resourceVersion: "460"
  selfLink: /api/v1/namespaces/kube-system/configmaps/coredns
  uid: be64aaad-1629-441f-8a40-a3efc0db9fa9
```

Después de modificar el archivo `hosts` en CoreDNS, no es necesario configurar el archivo `hosts` en cada pod.

----Fin

## Adición de la configuración Rewrite de CoreDNS para apuntar el nombre de dominio a los servicios en el clúster

Utilice el complemento Rewrite de CoreDNS para resolver un nombre de dominio especificado en el nombre de dominio de un Service.

**Paso 1** Utilice kubectl para conectarse al clúster.

**Paso 2** Modifique el archivo de configuración de CoreDNS para que apunte **example.com** al servicio **example** en el espacio de nombres **default**.

```
$ kubectl edit configmap coredns -n kube-system
apiVersion: v1
data:
  Corefile: |-
    .:5353 {
      bind {$POD_IP}
      cache 30
      errors
      health {$POD_IP}:8080
      kubernetes cluster.local in-addr.arpa ip6.arpa {
        pods insecure
        fallthrough in-addr.arpa ip6.arpa
      }
      rewrite name example.com example.default.svc.cluster.local
      loadbalance round_robin
      prometheus {$POD_IP}:9153
      forward . /etc/resolv.conf
      reload
    }
kind: ConfigMap
metadata:
  creationTimestamp: "2021-08-23T13:27:28Z"
  labels:
    app: coredns
    k8s-app: coredns
    kubernetes.io/cluster-service: "true"
    kubernetes.io/name: CoreDNS
    release: cceaddon-coredns
  name: coredns
  namespace: kube-system
  resourceVersion: "460"
  selfLink: /api/v1/namespaces/kube-system/configmaps/coredns
  uid: be64aaad-1629-441f-8a40-a3efc0db9fa9
```

----Fin

## Uso de CoreDNS a DNS autoconstruido en cascada

**Paso 1** Utilice kubectl para conectarse al clúster.

**Paso 2** Modifique el archivo de configuración de CoreDNS y cambie **/etc/resolv.conf** después de **forward** a la dirección IP del servidor de DNS externo.

```
$ kubectl edit configmap coredns -n kube-system
apiVersion: v1
data:
  Corefile: |-
    .:5353 {
      bind {$POD_IP}
      cache 30
      errors
      health {$POD_IP}:8080
      kubernetes cluster.local in-addr.arpa ip6.arpa {
        pods insecure
        fallthrough in-addr.arpa ip6.arpa
      }
    }
```

```
        loadbalance round_robin
        prometheus {$POD_IP}:9153
        forward . 192.168.1.1
        reload
    }
kind: ConfigMap
metadata:
  creationTimestamp: "2021-08-23T13:27:28Z"
  labels:
    app: coredns
    k8s-app: coredns
    kubernetes.io/cluster-service: "true"
    kubernetes.io/name: CoreDNS
    release: cceaddon-coredns
  name: coredns
  namespace: kube-system
  resourceVersion: "460"
  selfLink: /api/v1/namespaces/kube-system/configmaps/coredns
  uid: be64aaad-1629-441f-8a40-a3efc0db9fa9
```

----Fin

## 7.5.4 Uso de DNSCache de NodeLocal para mejorar el rendimiento de DNS

### Desafíos

Durante la resolución de DNS, si hay un gran número de solicitudes, CoreDNS estará bajo presión, lo que tiene los siguientes impactos:

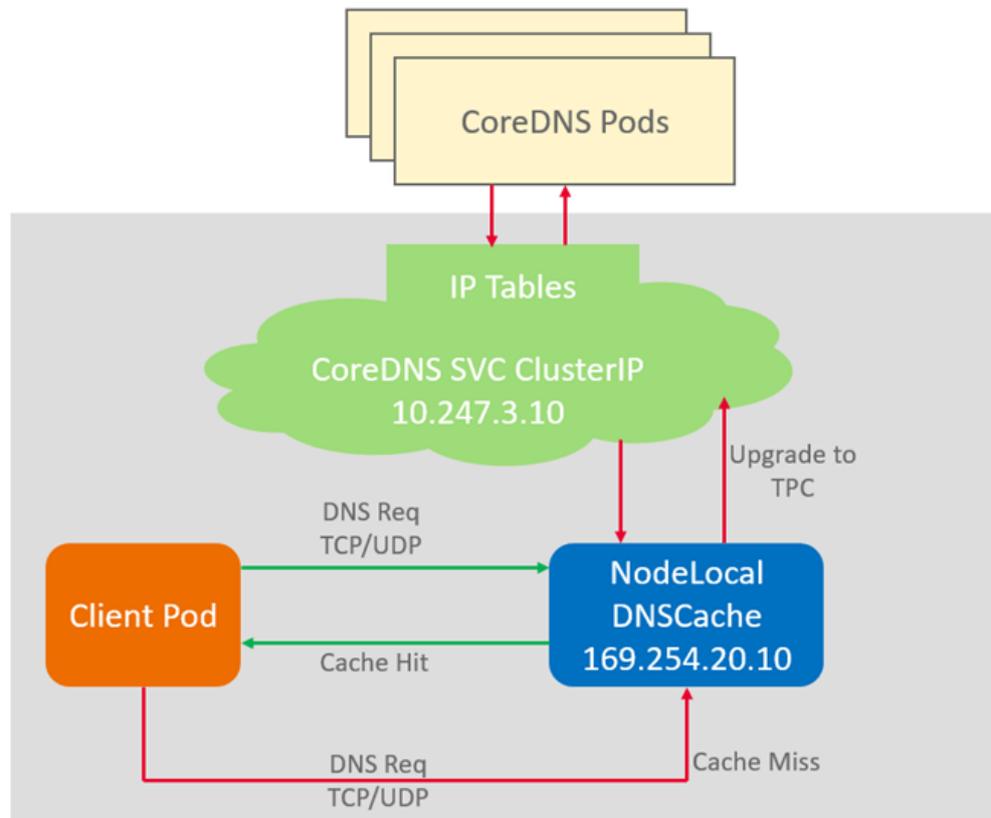
- La consulta se vuelve lenta, lo que afecta al rendimiento del servicio.
- CoreDNS requiere especificaciones más altas.

### Solución

DNSCache de NodeLocal mejora el rendimiento de DNS del clúster mediante la ejecución de proxys de caché de DNS en nodos del clúster.

Después de habilitar DNSCache de NodeLocal, una consulta de DNS pasa por la ruta como se muestra a continuación.

**Figura 7-33** Ruta de consulta de NodeLocal DNSCache



## Instalación del complemento

CCE proporciona **node-local-dns** adicional para que instale DNSCache de NodeLocal.

### 📖 NOTA

- El complemento node-local-dns solo admite clústeres de v1.19 y posteriores.
- DNSCache de NodeLocal sirve como un proxy de almacenamiento en caché transparente para CoreDNS y no proporciona complementos como hosts o reescritura. Si desea habilitar estos complementos, modifique las configuraciones de CoreDNS.
- Los pods no se pueden inyectar automáticamente en el espacio de nombres del kube-system.

**Paso 1** (Opcional) Modifique la configuración de CoreDNS para que CoreDNS utilice preferentemente UDP para comunicarse con el servidor DNS ascendente.

El DNSCache de NodeLocal utiliza TCP para comunicarse con el CoreDNS. El CoreDNS se comunica con el servidor DNS ascendente basándose en el protocolo utilizado por el origen de la solicitud. Sin embargo, el servidor en la nube no es compatible con TCP. Para utilizar DNSCache de NodeLocal, debe modificar la configuración de CoreDNS para que UDP se utilice preferentemente para comunicarse con el servidor de DNS ascendente, evitando excepciones de resolución.

Ejecute el siguiente comando:

```
kubectl edit configmap coredns -nkube-system
```

En el complemento de reenvío, especifique **prefer\_udp** como el protocolo utilizado por las solicitudes. Después de la modificación, CoreDNS utiliza preferentemente UDP para comunicarse con el sistema aguas arriba.

```
forward . /etc/resolv.conf { prefer_udp }
```

**Paso 2** Inicie sesión en la consola de CCE y acceda a la consola del clúster. Elija **Add-ons** en el panel de navegación, localice **node-local-dns** a la derecha y haga clic en **Install**.

**Paso 3** En la página **Install Add-on**, seleccione las especificaciones del complemento y establezca los parámetros relacionados.

**enable\_dnsconfig\_admission**: si se debe inyectar automáticamente DNSConfig a los pods recién creados. El valor predeterminado es **false**. El valor **true** permite la inyección automática, evitando la inyección de archivos YAML pod configurados manualmente.

**Paso 4** Haga clic en **Install**.

----Fin

## Uso de DNSCache de NodeLocal

Puede utilizar DNSCache de NodeLocal de cualquiera de las siguientes maneras:

- Inyección automática: configure automáticamente el campo **dnsConfig** del pod al crearlo. (Los pods no se pueden inyectar automáticamente en el espacio de nombres del kube-system.)
- Configuración manual: configure manualmente el campo **dnsConfig** del pod.

### Inyección automática

Deben cumplirse las siguientes condiciones:

- **enable\_dnsconfig\_admission** se ha establecido en **true** para el complemento.
- La etiqueta **node-local-dns-injection=enabled** se ha agregado al espacio de nombres.  
**kubectl label namespace default node-local-dns-injection=enabled**
- El nuevo pod no se ejecuta en el espacio de nombres de kube-system o de kube-public.
- La etiqueta **node-local-dns-injection=disabled** para deshabilitar la inyección de DNS no se agrega al nuevo pod.
- El nuevo pod utiliza la red host y **DNSPolicy** es **ClusterFirstWithHostNet**. Alternativamente, el pod no utiliza la red anfitriona y **DNSPolicy** es **ClusterFirst**.

Después de activar la inyección automática, los siguientes ajustes de **dnsConfig** se agregan automáticamente al pod creado. Además de la dirección de DNSCache de NodeLocal 169.254.20.10, la dirección de CoreDNS 10.247.3.10 se agrega a **nameservers** asegurando una alta disponibilidad del servidor de DNS del servicio.

```
dnsConfig:
  nameservers:
    - 169.254.20.10
    - 10.247.3.10
  searches:
    - default.svc.cluster.local
    - svc.cluster.local
    - cluster.local
  options:
    - name: timeout
      value: ''
    - name: ndots
```

```
value: '5'  
- name: single-request-reopen
```

### Configuración manual

Agregue manualmente la configuración **dnsConfig** al pod.

Cree un pod y configure **dnsConfig** en **169.254.20.10**.

```
apiVersion: v1  
kind: Pod  
metadata:  
  name: nginx  
spec:  
  containers:  
  - image: nginx:alpine  
    name: container-0  
  dnsConfig:  
    nameservers:  
    - 169.254.20.10  
    searches:  
    - default.svc.cluster.local  
    - svc.cluster.local  
    - cluster.local  
    options:  
    - name: ndots  
      value: '2'  
  imagePullSecrets:  
  - name: default-secret
```

## 7.6 Configuración de red de contenedores

### 7.6.1 Red de host

#### Escenario

Kubernetes permite que los pods utilicen directamente la red host/node. Cuando un pod se configura con **hostNetwork: true**, las aplicaciones que se ejecutan en el pod pueden ver directamente la interfaz de red del host donde se encuentra el pod.

#### Configuración

Agregue **hostNetwork: true** a la definición de pod.

```
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  name: nginx  
spec:  
  replicas: 1  
  selector:  
    matchLabels:  
      app: nginx  
  template:  
    metadata:  
      labels:  
        app: nginx  
    spec:  
      hostNetwork: true  
      containers:  
      - image: nginx:alpine  
        name: nginx
```

```
imagePullSecrets:
- name: default-secret
```

La configuración se realiza correctamente si la IP del pod es la misma que la IP del nodo.

```
$ kubectl get pod -owide
NAME                                READY    STATUS    RESTARTS   AGE    IP
NODE                                NOMINATED NODE  READINESS GATES
nginx-6fdf99c8b-6wwft              1/1     Running   0          3m41s  10.1.0.55
10.1.0.55 <none>                   <none>
```

## Precauciones

Si un pod utiliza la red host, ocupa un puerto host. La IP del pod es la IP del host. Para utilizar la red host, debe confirmar que los pods no entran en conflicto entre sí en términos de los puertos host que ocupan. No utilice la red host a menos que sepa exactamente qué puerto host utiliza cada pod.

Cuando se utiliza la red host, se accede al nodo para acceder a un pod en él. Por lo tanto, es necesario **permitir el acceso desde el puerto del grupo de seguridad del nodo**. De lo contrario, el acceso falla.

Además, el uso de la red host requiere que reserve puertos host para los pods. Cuando utilice una Deployment para desplegar pods del tipo hostNetwork, asegúrese de que **el número de pods no excede el número de nodos**. De lo contrario, se programarán varios pods en el nodo y no se iniciarán debido a conflictos de puertos. Por ejemplo, en el ejemplo anterior de nginx YAML, si se despliegan dos pods (configurando **replicas** a **2**) en un clúster con un solo nodo, no se puede crear un pod. Los logs de pod mostrarán que el Nginx no se puede iniciar porque el puerto está ocupado.

### ATENCIÓN

No programe varios pods que usen la red host en el mismo nodo. De lo contrario, cuando se crea un Service de ClusterIP para tener acceso a un pod, no se puede tener acceso a la dirección IP del clúster.

```
$ kubectl get deploy
NAME    READY    UP-TO-DATE    AVAILABLE    AGE
nginx  1/2      2              1            67m
$ kubectl get pod
NAME                                READY    STATUS                    RESTARTS   AGE
nginx-6fdf99c8b-6wwft              1/1     Running                  0          67m
nginx-6fdf99c8b-rglm7              0/1     CrashLoopBackOff        13         44m
$ kubectl logs nginx-6fdf99c8b-rglm7
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to
perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-
default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/
default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/
conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2022/05/11 07:18:11 [emerg] 1#1: bind() to 0.0.0.0:80 failed (98: Address in use)
nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address in use)
2022/05/11 07:18:11 [emerg] 1#1: bind() to [::]:80 failed (98: Address in use)
nginx: [emerg] bind() to [::]:80 failed (98: Address in use)
2022/05/11 07:18:11 [emerg] 1#1: bind() to 0.0.0.0:80 failed (98: Address in use)
```

```

nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address in use)
2022/05/11 07:18:11 [emerg] 1#1: bind() to [::]:80 failed (98: Address in use)
nginx: [emerg] bind() to [::]:80 failed (98: Address in use)
2022/05/11 07:18:11 [emerg] 1#1: bind() to 0.0.0.0:80 failed (98: Address in use)
nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address in use)
2022/05/11 07:18:11 [emerg] 1#1: bind() to [::]:80 failed (98: Address in use)
nginx: [emerg] bind() to [::]:80 failed (98: Address in use)
2022/05/11 07:18:11 [emerg] 1#1: bind() to 0.0.0.0:80 failed (98: Address in use)
nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address in use)
2022/05/11 07:18:11 [emerg] 1#1: bind() to [::]:80 failed (98: Address in use)
nginx: [emerg] bind() to [::]:80 failed (98: Address in use)
2022/05/11 07:18:11 [emerg] 1#1: bind() to 0.0.0.0:80 failed (98: Address in use)
nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address in use)
2022/05/11 07:18:11 [emerg] 1#1: bind() to [::]:80 failed (98: Address in use)
nginx: [emerg] bind() to [::]:80 failed (98: Address in use)
2022/05/11 07:18:11 [emerg] 1#1: still could not bind()
nginx: [emerg] still could not bind()
    
```

## 7.6.2 Configuración de la limitación de la velocidad de QoS para el acceso entre los pod

### Escenario

La preferencia de ancho de banda se produce entre diferentes contenedores desplegados en el mismo nodo, lo que puede causar fluctuación de servicio. Puede configurar la limitación de la velocidad de QoS para el acceso entre los pod para evitar este problema.

### Restricciones

A continuación se muestran las restricciones al establecer la limitación de velocidad para el acceso entre los pod:

Tipo de restricción	Modelo de red de túneles	Modelo de red de VPC	Modelo de la red de Cloud Native 2.0
Versiones compatibles	Todas las versiones	Clústeres de v1.19.10 y posteriores	Clústeres de v1.19.10 y posteriores
Tipos de tiempo de ejecución admitidos	Solo se admiten contenedores comunes (runC como el tiempo de ejecución contenedor). No se admiten contenedores seguros.	Solo se admiten contenedores comunes (runC como el tiempo de ejecución contenedor). No se admiten contenedores seguros (Kata como el tiempo de ejecución contenedor).	Solo se admiten contenedores comunes (runC como el tiempo de ejecución contenedor). No se admiten contenedores seguros (Kata como el tiempo de ejecución contenedor).
Tipos de pod admitidos	Solo los pod que no sean de HostNetwork		

Tipo de restricción	Modelo de red de túneles	Modelo de red de VPC	Modelo de la red de Cloud Native 2.0
Escenarios soportados	Acceso entre los pod, pod que acceden a nodos y pod que acceden a servicios		
Restricciones	No hay	No hay	<ul style="list-style-type: none"> <li>● Los pods acceden a los bloques CIDR del servicio en la nube externo 100.64.0.0/10 y 214.0.0.0/8.</li> <li>● Limitación de la tasa de tráfico del control de salud</li> </ul>
Límite de tasa superior	Valor mínimo entre el límite de ancho de banda superior y 34 Gbit/s	Valor mínimo entre el límite de ancho de banda superior y 4.3 Gbit/s	Valor mínimo entre el límite de ancho de banda superior y 4.3 Gbit/s
Límite de velocidad más bajo	Solo se admite el límite de velocidad de Kbit/s o superior.	Actualmente, solo se admite el límite de velocidad de Mbit/s o superior.	

## Uso de kubectl

Puede agregar anotaciones a una carga de trabajo para especificar su ancho de banda de salida y entrada.

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: test
  namespace: default
  labels:
    app: test
spec:
  replicas: 2
  selector:
    matchLabels:
      app: test
  template:
    metadata:
      labels:
        app: test
      annotations:
        kubernetes.io/ingress-bandwidth: 100M
        kubernetes.io/egress-bandwidth: 100M
    spec:
      containers:
        - name: container-1
    
```

```
image: nginx:alpine
imagePullPolicy: IfNotPresent
imagePullSecrets:
- name: default-secret
```

- **kubernetes.io/ingress-bandwidth**: ancho de banda de entrada del pod
- **kubernetes.io/egress-bandwidth**: ancho de banda de salida del pod

Si no se especifican estos dos parámetros, el ancho de banda no está limitado.

#### NOTA

Después de modificar el límite de ancho de banda de entrada o salida de un pod, debe reiniciar el contenedor para que la modificación surta efecto. Después de modificar las anotaciones en un pod no gestionado por cargas de trabajo, el contenedor no se reiniciará, por lo que los límites de ancho de banda no surten efecto. Puede volver a crear un pod o reiniciar el contenedor manualmente.

## 7.6.3 Configuración de la red del túnel del contenedor

### 7.6.3.1 Network Policies

NetworkPolicy es un objeto de Kubernetes que se utiliza para restringir el acceso a pods. En CCE, al establecer las políticas de red, puede definir las reglas de entrada especificando las direcciones a las que acceder los pods o las reglas de salida especificando las direcciones a las que pueden acceder los pods. Esto equivale a configurar un firewall en la capa de aplicación para garantizar aún más la seguridad de la red.

Las políticas de red dependen del complemento de red del clúster al que se aplican las políticas.

De forma predeterminada, si un espacio de nombres no tiene ninguna política, los pods del espacio de nombres aceptan tráfico de cualquier origen y envían tráfico a cualquier destino.

Las reglas de política de red se clasifican en los siguientes tipos:

- **namespaceSelector**: selecciona los espacios de nombres particulares para los que se deben permitir todos los pods como fuentes de entrada o destinos de salida.
- **podSelector**: selecciona los pods particulares en el mismo espacio de nombres que la política de red que debe permitirse como fuentes de entrada o destinos de salida.
- **ipBlock**: selecciona los bloques IP particulares para permitirlos como fuentes de entrada o destinos de salida.

### Notas y restricciones

- Solo los clústeres que utilizan el modelo de red de túnel admiten políticas de red.
- Las salidas no son compatibles con las políticas de red.
- El aislamiento de red no es compatible con las direcciones IPv6.

### Uso de reglas de ingreso

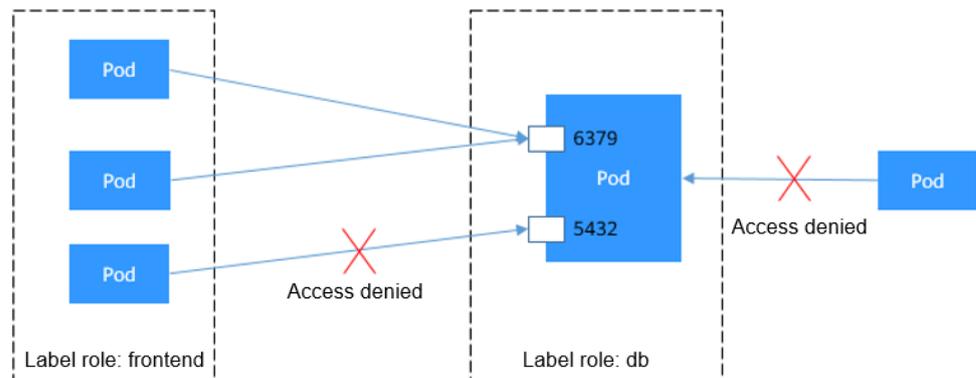
- **Uso de podSelector para especificar el ámbito de acceso**

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: test-network-policy
  namespace: default
```

```
spec:
  podSelector:
    # The rule takes effect for pods with the
    role=db label.
    matchLabels:
      role: db
  ingress:
    # This is an ingress rule.
    - from:
      - podSelector:
          # Only traffic from the pods with the
          "role=frontend" label is allowed.
          matchLabels:
            role: frontend
      ports:
        # Only TCP can be used to access port 6379.
        - protocol: TCP
          port: 6379
```

La siguiente figura muestra cómo funciona podSelector.

**Figura 7-34** podSelector

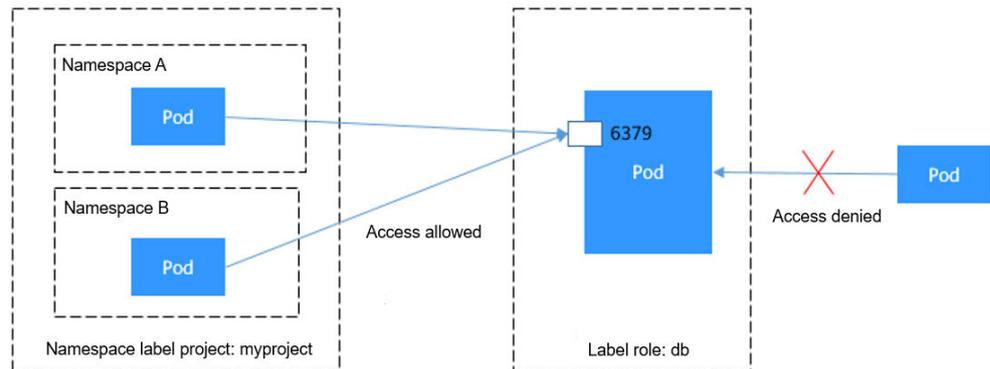


● **Uso de namespaceSelector para especificar el ámbito de acceso**

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: test-network-policy
spec:
  podSelector:
    # The rule takes effect for pods with the
    role=db label.
    matchLabels:
      role: db
  ingress:
    # This is an ingress rule.
    - from:
      - namespaceSelector:
          # Only traffic from the pods in the namespace
          with the "project=myproject" label is allowed.
          matchLabels:
            project: myproject
      ports:
        # Only TCP can be used to access port 6379.
        - protocol: TCP
          port: 6379
```

La siguiente figura muestra cómo funciona namespaceSelector.

**Figura 7-35 namespaceSelector**



## Creación de una política de red en la consola

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** Elija **Networking** en el panel de navegación, haga clic en la ficha **Network Policies** y haga clic en **Create Network Policy** en la esquina superior derecha.

- **Policy Name:** Especifique un nombre de política de red.
- **Namespace:** seleccione un espacio de nombres en el que se aplique la política de red.
- **Selector:** Ingrese una etiqueta, seleccione el pod a asociar y haga clic en **Add**. También puede hacer clic en **Reference Workload Label** para hacer referencia a la etiqueta de una carga de trabajo existente.
- **Inbound Rule:** Haga clic en **+** para agregar una regla entrante. Para obtener más información sobre la configuración de los parámetros, consulte [Tabla 7-41](#).

Inbound Rule	Policies	Protocol & Port ?	Source Namespace ?	Source Pod Label ?	Operation
Allow	TCP	80	default	app = nginx-test version = v1	Delete

**Tabla 7-41** Adición de una regla de entrada

Parámetro	Descripción
Protocol & Port	Seleccione el tipo de protocolo y el puerto. Actualmente, se soportan TCP y UDP.
Source Namespace	Seleccione un espacio de nombres a cuyos objetos se puede acceder. Si no se especifica este parámetro, el objeto pertenece al mismo espacio de nombres que la política actual.
Source Pod Label	Permitir el acceso a los pods con esta etiqueta. Si no se especifica este parámetro, se puede acceder a todos los pods del espacio de nombres.

**Paso 3** Haga clic en **OK**.

----Fin

## 7.6.4 Configuración de Cloud Native Network 2.0

### 7.6.4.1 Políticas de grupo de seguridad

Cuando se utiliza el modelo de Cloud Native Network 2.0, los pods utilizan ENI de VPC o subENI para redes. Puede vincular directamente los grupos de seguridad y las EIP a los pods. CCE proporciona un objeto de recurso personalizado denominado **SecurityGroup** para que pueda asociar grupos de seguridad con pods en CCE. SecurityGroups puede personalizar cargas de trabajo con requisitos específicos de aislamiento de seguridad.

### Restricciones

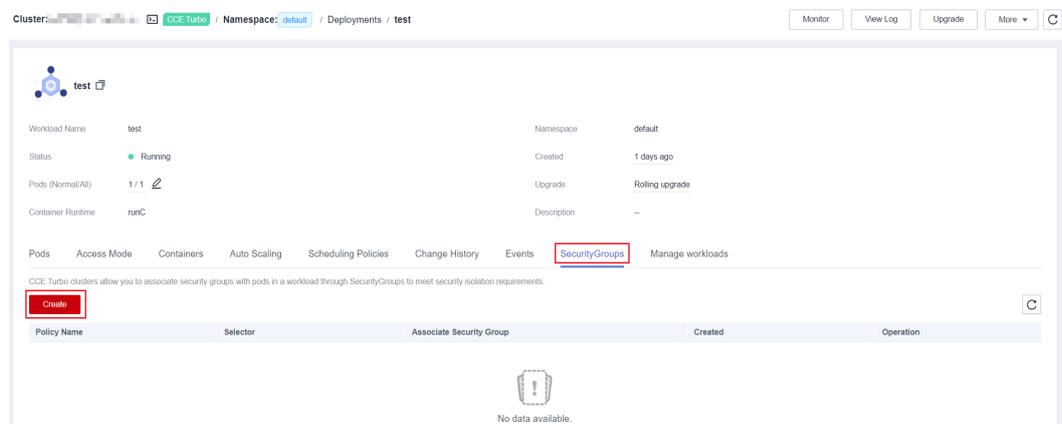
- Esta función es compatible con los clústeres de CCE Turbo de v1.19 y posteriores. Actualice sus clústeres de CCE Turbo si sus versiones son anteriores a v1.19.
- Una carga de trabajo puede vincularse a un máximo de cinco grupos de seguridad.

### Uso de la consola

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** En el panel de navegación, elija **Workloads**. En la página mostrada, haga clic en el nombre de la carga de trabajo de destino.

**Paso 3** Cambie a la página de la ficha **Security Group Policy** y haga clic en **Create**.



**Paso 4** Establezca los parámetros como se describe en [Tabla 7-42](#).

**Tabla 7-42** Parámetros de configuración

Parámetro	Descripción	Valor de ejemplo
Security Group Policy Name	<p>Escriba un nombre de política de seguridad.</p> <p>Introduzca de 1 a 63 caracteres. El valor debe comenzar con una minúscula y no puede finalizar con un guion (-). Solo se permiten letras minúsculas, dígitos y guiones (-).</p>	security-group
Associate Security Group	<p>El grupo de seguridad seleccionado estará vinculado a la ENI o a la ENI suplementaria de la carga de trabajo seleccionada. Se puede seleccionar un máximo de cinco grupos de seguridad de la lista desplegable. Para crear un SecurityGroup debe seleccionar uno o varios grupos de seguridad.</p> <p>Si no se ha creado ningún grupo de seguridad, haga clic en <b>Create Security Group</b>. Una vez creado el grupo de seguridad, haga clic en el botón de actualizar.</p> <p><b>AVISO</b></p> <ul style="list-style-type: none"> <li>● Se puede seleccionar un máximo de 5 grupos de seguridad.</li> <li>● Sitúe el cursor sobre  junto al nombre del grupo de seguridad y podrá ver detalles sobre el grupo de seguridad.</li> </ul>	<p>64566556-bd6f-48fb-b2c6-df8f44617953</p> <p>5451f1b0-bd6f-48fb-b2c6-df8f44617953</p>

**Paso 5** Después de establecer los parámetros, haga clic en **OK**.

Después de crear el grupo de seguridad, el sistema vuelve automáticamente a la página de lista de grupos de seguridad, donde puede ver el nuevo grupo de seguridad.

----Fin

## Uso de kubectl

**Paso 1** Utilice kubectl para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).

**Paso 2** Cree un archivo de descripción denominado **securitygroup-demo.yaml**.

**vi securitygroup-demo.yaml**

Por ejemplo, cree el siguiente SecurityGroup para enlazar todas las cargas de trabajo nginx con dos grupos de seguridad 64566556-bd6f-48fb-b2c6-df8f44617953 y 5451f1b0-bd6f-48fb-b2c6-df8f44617953 que se han creado de antemano. Un ejemplo es el siguiente:

```

apiVersion: crd.yangtse.cni/v1
kind: SecurityGroup
metadata:
  name: demo
  namespace: default
spec:
  podSelector:
    matchLabels:
      app: nginx
  securityGroups:
    - id: 64566556-bd6f-48fb-b2c6-df8f44617953
    - id: 5451f1b0-bd6f-48fb-b2c6-df8f44617953
    
```

**Tabla 7-43** describe los parámetros del archivo YAML.

**Tabla 7-43** Descripción

Campo	Descripción	Obligatorio
apiVersion	Versión de la API. El valor es <b>crd.yangtse.cni/v1</b> .	Sí
kind	Tipo del objeto que se va a crear.	Sí
metadata	Definición de metadatos del objeto de recurso.	Sí
name	Nombre del SecurityGroup.	Sí
namespace	Nombre del espacio de nombres.	Sí
spec	Descripción detallada del SecurityGroup.	Sí
podSelector	Se utiliza para definir la carga de trabajo que se asociará a los grupos de seguridad en SecurityGroup.	Sí
securityGroups	ID del grupo de seguridad.	Sí

**Paso 3** Ejecute el siguiente comando para crear el SecurityGroup:

```
kubectl create -f securitygroup-demo.yaml
```

Si se muestra la siguiente información, se está creando el SecurityGroup.

```
securitygroup.crd.yangtse.cni/demo created
```

**Paso 4** Ejecute el siguiente comando para ver el SecurityGroup:

```
kubectl get sg
```

Si el nombre del SecurityGroup creado es **demo** en la salida del comando, el SecurityGroup se crea correctamente.

```

NAME          POD-SELECTOR          AGE
all-no       map[matchLabels:map[app:nginx]]  4h1m
s001test     map[matchLabels:map[app:nginx]]  19m
demo         map[matchLabels:map[app:nginx]]  2m9s
    
```

----Fin

## 7.7 Configuración de red de clúster

### 7.7.1 Adición de un bloque CIDR de VPC secundario para un clúster

#### Escenario

Al crear un clúster, debe desplegarlo en una VPC. Si la VPC planificada es demasiado pequeña y las direcciones IP son insuficientes, puede usar un bloque CIDR de VPC secundario para admitir el ajuste de su servicio. Esta sección describe cómo agregar un bloque CIDR de VPC secundario para su clúster.

#### Notas y restricciones

Solo se admiten los clústeres de CCE y de CCE Turbo de v1.21 y posteriores.

#### Planificación de un bloque secundario CIDR

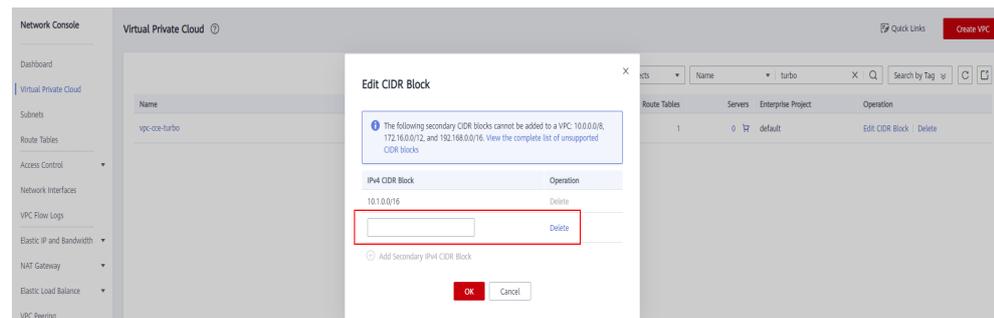
Antes de agregar un bloque secundario de CIDR, planifique adecuadamente para prevenir los conflictos de CIDR. Tener en cuenta los siguientes puntos:

1. Todas las subredes (incluidas las subredes extendidas) de la VPC donde reside el clúster no pueden entrar en conflicto con los bloques contenedor y Service CIDR.
2. Los bloques de CIDR 10.0.0.0/8, 172.16.0.0/12 y 192.168.0.0/16 pueden entrar en conflicto con las direcciones IP asignadas a los nodos maestros del clúster. No se recomienda seleccionarlos como bloques CIDR secundarios.
3. Si un ECS que no está en un clúster en la misma VPC necesita tener acceso al clúster, se realiza la Secure Network Address Translation (SNAT). La dirección de origen de pod es la dirección IP del nodo en lugar de la dirección IP de pod.
4. Los ECS de un bloque CIDR secundario no pueden acceder a los pods del clúster a menos que este bloque CIDR se haya utilizado para agregar nodos en el clúster.

#### Procedimiento

**Paso 1** Inicie sesión en la consola de VPC, seleccione la VPC a la que pertenece el clúster, haga clic en **Edit CIDR Block** y haga clic en **Add Secondary IPv4 CIDR Block**.

**Figura 7-36** Adición de un bloque CIDR IPv4 secundario



**Paso 2** Cree una subred en el bloque CIDR IPv4 secundario para el clúster.

**Create Subnet** ×

\* VPC  C  
IPv4 CIDR block: 10.1.0.0/16  
The VPC already contains 2 subnets.

\* AZ  ?

\* Name

\* IPv4 CIDR Block  C  
Available IP Addresses: 251  
The CIDR block cannot be modified after the subnet has been created.

IPv6 CIDR Block  Enable ?

Associated Route Table  ?

Advanced Settings ▼ Gateway | DNS Server Address | NTP Server Address |  
DHCP Lease Time | Tag | Description

**Paso 3** Después de crear una subred utilizando el bloque CIDR IPv4 secundario, puede seleccionar la subred al crear un nodo o un grupo de nodos en la página **Network Settings**.

**Network Settings** Configure networking resources for node and application communication.

VPC

Node Subnet  C Available Subnet IP Addresses: 241

🔔 If the default DNS server of the subnet is modified, ensure that the custom DNS server can resolve the OBS service domain name. Otherwise, the node cannot be created.

Node IP

EIP    ?

----Fin

## 7.7.2 Cambio de una subred de nodo

### Escenario

En esta sección se describe cómo cambiar subredes para nodos de un clúster.

## Restricciones

- Solo se pueden conmutar subredes en la misma VPC que el clúster. No se puede cambiar el grupo de seguridad del nodo.
- Al cambiar la subred de un nodo, cumpla con las restricciones en [Cambio de una VPC](#).

## Procedimiento

**Paso 1** Inicie sesión en la consola de ECS.

**Paso 2** Haga clic en **More > Manage Network > Change VPC** en la columna **Operation** del ECS de destino.

**Paso 3** Establezca los parámetros para cambiar la VPC.

- **VPC:** Seleccione la misma VPC que la del clúster.
- **Subnet:** Seleccione la subred de destino que se va a cambiar.
- **Private IP Address:** Seleccione **Assign new** o **Use existing** según sea necesario.
- **Security Group:** Seleccione el grupo de seguridad del nodo del clúster. De lo contrario, el nodo no está disponible.

×

### Change VPC

Changing the VPC will interrupt ECS network connections and change the subnet, IP address, and MAC address of the ECS.  
During the change process, do not perform operations on the ECS, including its EIP.  
After the VPC is changed, to ensure services are not impacted, reconfigure source/destination check, virtual IP address, and network-related application software and services, such as ELB, VPN, NAT, and DNS.

ECS Name ecs-xxxxxx

VPC vpc-cce(192.168.0.0/16) View In-Use VPCs

Subnet cci-subnet-xoimk5(192.168.32.0/19) View Subnet

Private IP Address Assign new Use existing

View In-Use IP Address

Security Group cce-test-cce-node-hd6nf View Security Group

OK Cancel

**Paso 4** Haga clic en **OK**.

**Paso 5** Vaya a la consola de CCE y restablezca el nodo. Puede utilizar la configuración de parámetros predeterminada. Para obtener más información, consulte [Restablecer un nodo](#).

----Fin

## 7.7.3 Adición de un bloque CIDR de contenedor para un clúster

### Escenario

Si el bloque CIDR contenedor (subred de contenedor en un clúster de CCE Turbo) establecido durante la creación del clúster CCE es insuficiente, puede agregar un bloque CIDR contenedor para el clúster.

### Restricciones

- Esta función es aplicable a los clústeres de CCE y de CCE Turbo de 1.19 o posterior, pero no es aplicable a los clústeres que utilizan el modelo de red de túneles contenedor.
- El bloque CIDR de contenedor o la subred contenedor no se pueden eliminar después de agregarse. Tenga cuidado al realizar esta operación.
- El bloque CIDR de servicio predeterminado es 10.247.0.0/16. Por lo tanto, el bloque CIDR del contenedor no puede ser 10.247.0.0/16.

### Adición de un bloque CIDR de contenedor para un clúster de CCE

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster de CCE.

**Paso 2** En la página **Cluster Information**, busque el área **Networking Configuration** y haga clic en **Add Pod Subnet**.

#### Networking Configuration

Network Model	VPC network
VPC	<a href="#">vpc-cce</a> 
Subnet	<a href="#">subnet-cce</a>
Container CIDR Block	10.0.0.0/16
	<a href="#">Add Container CIDR Block</a>
IPv4 Service CIDR Block	10.247.0.0/16
Forwarding	iptables
Default Node Security Group	<a href="#">cce-test-cce-node-hd6nf</a>  

**Paso 3** Configure el bloque CIDR de contenedor que se agregará. Puede hacer clic en  para agregar varios bloques CIDR de contenedor a la vez.

## Add Container CIDR Block



The added container CIDR block cannot be deleted.

Container  .  .  .  /

CIDR Block

Max. nodes supported by the current networking configuration: 1,024

OK

Cancel

**Paso 4** Haga clic en **OK**.

----Fin

## Adición de una subred de contenedores para un clúster de CCE Turbo

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster de CCE Turbo.

**Paso 2** En la página **Cluster Information**, busque el área **Networking Configuration** y haga clic en **Add Pod Subnet**.

### Networking Configuration

Network Model Cloud Native Network 2.0

VPC [vpc-cidr](#)

Subnet [subnet-a008](#)

Default Pod Subnet [subnet-cfc8](#)

**Add Pod Subnet**

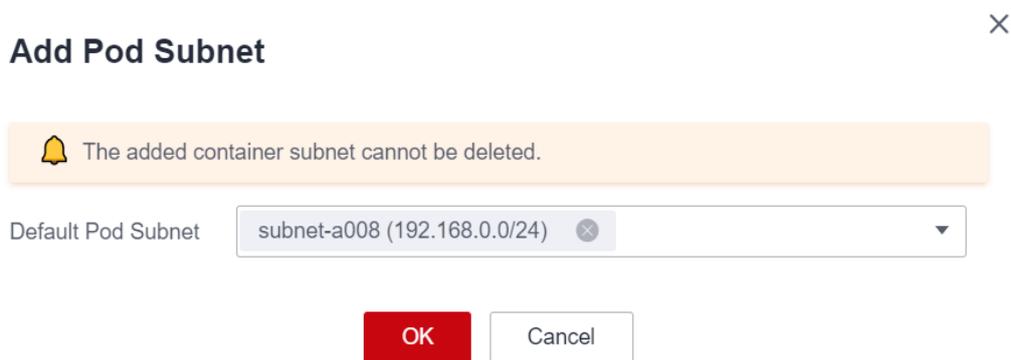
[goto Network Configuration](#)

IPv4 Service CIDR Block 10.247.0.0/16

Forwarding iptables

Default Node Security Group [-cce-node-lq9cs](#)

**Paso 3** Seleccione una subred de contenedor en la misma VPC. Puede agregar varias subredes de contenedor a la vez. Si no hay ninguna otra subred de contenedor disponible, vaya a la consola de VPC para crear una.



**Paso 4** Haga clic en **OK**.

----Fin

## 7.8 Configuración del acceso dentro de la VPC

En esta sección se describe cómo acceder a una intranet desde un contenedor (fuera del clúster en una VPC), incluido el acceso dentro de VPC y el acceso entre VPC.

### Acceso dentro de la VPC

El rendimiento de acceder a una intranet desde un contenedor varía según los modelos de red de contenedor de un clúster.

- **Container tunnel network**

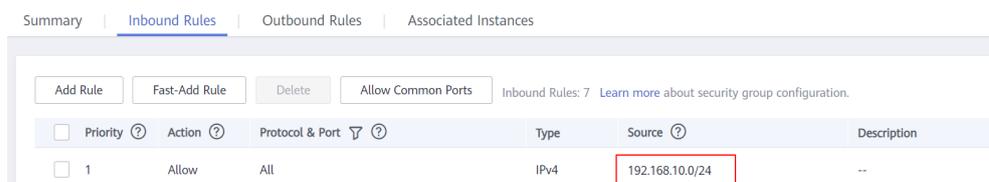
La red de túneles de contenedor encapsula paquetes de datos de red con túneles basados en la red de nodos. Un contenedor puede acceder a otros recursos en la misma VPC siempre que el nodo pueda acceder a los recursos. Si el acceso falla, compruebe si el grupo de seguridad del recurso del par permite el acceso desde el nodo donde se encuentra el contenedor.

- **VPC network**

El modelo de red de VPC utiliza rutas de VPC para reenviar el tráfico de contenedor. El bloque CIDR de contenedor y el nodo de VPC no están en el mismo bloque CIDR. Cuando un contenedor accede a otros recursos en la misma VPC, haga clic en **el grupo de seguridad del recurso del par debe permitir el acceso al bloque CIDR de contenedor**.

Por ejemplo, el bloque CIDR donde reside el nodo de clúster es 192.168.10.0/24, y el bloque CIDR de contenedor es 172.16.0.0/16.

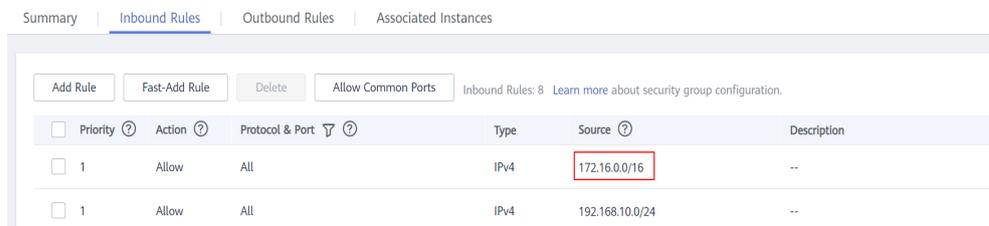
Hay un ECS cuya dirección IP es 192.168.10.52 en la VPC (fuera del clúster). El grupo de seguridad del ECS permite el acceso solo al bloque CIDR del nodo del clúster.



En este caso, si hace ping 192.168.10.52 desde contenedor, la operación de ping falla.

```
kubectl exec test01-6cbbf97b78-krj6h -it -- /bin/sh
/ # ping 192.168.10.25
PING 192.168.10.25 (192.168.10.25): 56 data bytes
^C
--- 192.168.10.25 ping statistics ---
104 packets transmitted, 0 packets received, 100% packet loss
```

Configure el grupo de seguridad para permitir el acceso desde el bloque CIDR de contenedor 172.16.0.0/16.



En este caso, se puede hacer un ping 192.168.10.52 desde el contenedor.

```
$ kubectl exec test01-6cbbf97b78-krj6h -it -- /bin/sh
/ # ping 192.168.10.25
PING 192.168.10.25 (192.168.10.25): 56 data bytes
64 bytes from 192.168.10.25: seq=0 ttl=64 time=1.412 ms
64 bytes from 192.168.10.25: seq=1 ttl=64 time=1.400 ms
64 bytes from 192.168.10.25: seq=2 ttl=64 time=1.299 ms
64 bytes from 192.168.10.25: seq=3 ttl=64 time=1.283 ms
^C
--- 192.168.10.25 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
```

## Acceso entre VPC

El acceso entre VPC se implementa estableciendo una interconexión entre las VPC.

- En el modelo de red de túnel de contenedor, un contenedor puede acceder al par de las VPC solo cuando se habilita la comunicación entre la red de nodo y el par de las VPC.
- Cloud Native Network 2.0 es similar a la red de túneles de contenedor. Solo necesita habilitar la comunicación entre la subred donde se encuentra el contenedor y la VPC del mismo nivel.
- Cada red de VPC tiene un bloque CIDR de contenedor independiente. Además del bloque CIDR de VPC, el bloque CIDR de contenedor también necesita estar conectado.

Suponga que hay dos VPC.

- vpc-demo: Su bloque CIDR es 192.168.0.0/16, el clúster está en vpc-demo, y el bloque CIDR de contenedor es 10.0.0.0/16.
- vpc-demo2: Su bloque CIDR es 10.1.0.0/16.

Crear una interconexión llamada **peering-demo** (la VPC local es vpc-demo y la VPC del otro extremo es vpc-demo2). Agregue el bloque CIDR de contenedor a la ruta de la VPC del mismo nivel.



Local Routes | **Peer Routes**

Switch to the [Route Tables](#) page to add routes for the VPC peering connection.

Destination	Next Hop Type	Next Hop
10.0.0.0/16	VPC peering connection	peering-demo(b42edde2-8084-4457-8b06-df8f1b1425eb)
192.168.0.0/16	VPC peering connection	peering-demo(b42edde2-8084-4457-8b06-df8f1b1425eb)

Después de esta configuración, puede acceder al bloque CIDR de contenedor 10.0.0.0/16 en vpc-demo2. Durante el acceso, preste atención a la configuración del grupo de seguridad y habilite la configuración del puerto.

## Acceso a otros servicios en la nube

Los servicios comunes que se comunican con CCE con una intranet incluyen RDS, DCS, Kafka, RabbitMQ y ModelArts.

Además de las configuraciones de red descritas en [Acceso dentro de la VPC](#) y [Acceso entre VPC](#) también debe comprobar la opción **si estos servicios en la nube permiten el acceso externo**. Por ejemplo, la instancia de DCS Redis solo puede ser accedida por las direcciones IP en su lista blanca. En general, se puede acceder a estos servicios en la nube mediante direcciones IP en la misma VPC. Sin embargo, el bloque CIDR de contenedor en el modelo de red de VPC es diferente del bloque CIDR de la VPC. Por lo tanto, debe agregar el bloque CIDR de contenedor a la lista blanca.

## ¿Qué pasa si un contenedor no accede a una intranet?

Si no se puede acceder a una intranet desde un contenedor, realice las siguientes operaciones:

1. Vea la regla de grupo de seguridad del servidor para comprobar si el contenedor tiene permiso para acceder al servidor del otro extremo.
  - El modelo de red de túnel de contenedor necesita permitir la dirección IP del nodo donde se encuentra el contenedor.
  - El modelo de red VPC necesita permitir el bloque CIDR de contenedor.
  - El modelo de Cloud Native Network 2.0 debe permitir la subred donde se encuentra el contenedor.
2. Compruebe si se ha configurado una lista blanca para el servidor del otro extremo. Por ejemplo, la instancia de DCS Redis solo puede ser accedida por las direcciones IP en su lista blanca. Agregue los bloques CIDR de contenedor y de nodo a la lista blanca.
3. Compruebe si el motor de contenedor está instalado en el servidor de otro extremo y si entra en conflicto con el bloque CIDR de contenedor en CCE. Si se produce un conflicto de red, el acceso falla.

## 7.9 Acceso a redes públicas desde un contenedor

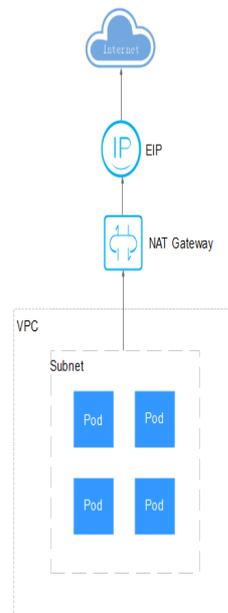
Los contenedores pueden acceder a las redes públicas de cualquiera de las siguientes maneras:

- Vincule una dirección IP pública al nodo donde se encuentra el contenedor si el modelo de red es la red de VPC o la red de túnel.
- Vincule una dirección IP pública al pod. (Cuando se utiliza el modelo de Cloud Native Network 2.0, vincule manualmente una EIP a la ENI o sub-ENI del pod en la consola de VPC. Este método no se recomienda porque la dirección IP de un pod cambia después de reprogramar el pod. Como resultado, el nuevo pod no puede acceder a la red pública.)

- Configurar reglas de SNAT con NAT Gateway.

Puede usar NAT Gateway para habilitar los pods de contenedor en una VPC para acceder a las redes públicas. NAT Gateway proporciona traducción de direcciones de red de origen (SNAT), que traduce las direcciones IP privadas a una dirección IP pública mediante la vinculación de una dirección IP elástica (EIP) al gateway, proporcionando un acceso seguro y eficiente a Internet. **Figura 7-37** muestra la arquitectura de SNAT. La función de SNAT permite que los pods de contenedor de una VPC accedan a Internet sin estar vinculados a una EIP. SNAT soporta un gran número de conexiones simultáneas, lo que lo hace adecuado para aplicaciones que implican un gran número de solicitudes y conexiones.

**Figura 7-37** SNAT



Para habilitar un pod de contenedor para acceder a Internet, realice los siguientes pasos:

**Paso 1** Asignar una EIP.

1. Inicie sesión en la consola de gestión.
2. Haga clic en  en la esquina superior izquierda de la consola de gestión y seleccione una región y un proyecto.
3. Haga clic en  en la esquina superior izquierda y elija **Networking** > **Elastic IP** en la lista expandida.
4. En la página **EIPs**, haga clic en **Buy EIP**.
5. Configure los parámetros según lo requerido.

 **NOTA**

Establezca **Region** en la región donde se encuentran los pods de contenedor.

**Figura 7-38** Comprar una dirección IP elástica

The screenshot shows the 'Buy EIP' configuration interface. At the top, there's a header with a back arrow and 'Buy EIP' with a help icon. Below this, the configuration is organized into sections:

- Billing Mode:** Two buttons, 'Yearly/Monthly' and 'Pay-per-use'.
- Region:** A dropdown menu showing 'CN East-Shanghai1'. Below it, a note states: 'An EIP can only be associated with cloud resources in the same region. The region cannot be changed after the EIP is purchased.'
- EIP Type:** Two buttons, 'Dynamic BGP' and 'Static BGP', with a help icon. Below them, a green checkmark indicates 'Greater than or equal to 99.95% service availability rate'.
- Billed By:** Three selectable options: 'Bandwidth' (For heavy/stable traffic), 'Traffic' (For light/sharply fluctuating traffic), and 'Shared Bandwidth' (For staggered traffic). Below this, it says 'Billed based on usage duration and bandwidth size.'
- Bandwidth:** A row of buttons for values 1, 2, 5, 10, 100, and 200, followed by a 'Custom' button and a numeric input field with '-' and '+' signs. A note says 'The value ranges from 1 to 2,000 Mbit/s'. Below this, a green checkmark indicates 'Free Anti-DDoS protection'.
- IPv6 EIP:** A checkbox for 'Enable IPv6 Internet access' with a help icon. Below it, a note says 'IPv6 EIP is free during the Open Beta Test.'
- Bandwidth Name:** A text input field containing 'bandwidth-6b17'.
- Enterprise Project:** A dropdown menu showing '--Select--' and a 'Create Enterprise Project' button with a help icon.

**Paso 2** Crear un gateway de NAT. Para obtener más información, consulte [Compra de un gateway de NAT](#).

1. Inicie sesión en la consola de gestión.
2. Haga clic en  en la esquina superior izquierda de la consola de gestión y seleccione una región y un proyecto.
3. Haga clic en  en la esquina superior izquierda y elija **Networking > NAT Gateway** en la lista expandida.
4. En la página mostrada, haga clic en **Buy Public NAT Gateway** en la esquina superior derecha.
5. Configure los parámetros según lo requerido.

 **NOTA**

Seleccione la misma VPC.

**Figura 7-39** Comprar un gateway de NAT

\* Billing Mode Yearly/Monthly Pay-per-use

\* Region 📍 CN East-Shanghai1

Regions are geographic areas isolated from each other. Resources are region-specific and cannot be used across regions th and quick resource access, select the nearest region.

\* Name

\* VPC my\_vpc 🔗 View VPC

\* Subnet my\_subnet (192.168.0.0/24) 🔗

The selected subnet is for the NAT gateway only. To enable communications over the Internet, after the NAT gateway is created, you need to add rules.

\* Type Small Medium Large Extra-large

Supports up to 10,000 connections. [Learn more](#)  
 The connections supported by a NAT gateway in a yearly/monthly subscription can always be increased later, but they cannot be decreased.

\* Enterprise Project default 🔗 Create Enterprise Project ?

**Paso 3** Configurar una regla de SNAT y vincular la EIP a la subred. Para obtener más información, consulte [Adición de una regla SNAT](#).

1. Inicie sesión en la consola de gestión.
2. Haga clic en  en la esquina superior izquierda de la consola de gestión y seleccione una región y un proyecto.
3. Haga clic en  en la esquina superior izquierda y elija **Networking** > **NAT Gateway** en la lista expandida.
4. En la página que se muestra, haga clic en el nombre del gateway de NAT para el que desea agregar la regla de SNAT.
5. En la página de ficha **SNAT Rules**, haga clic en **Add SNAT Rule**.
6. Configure los parámetros según lo requerido.

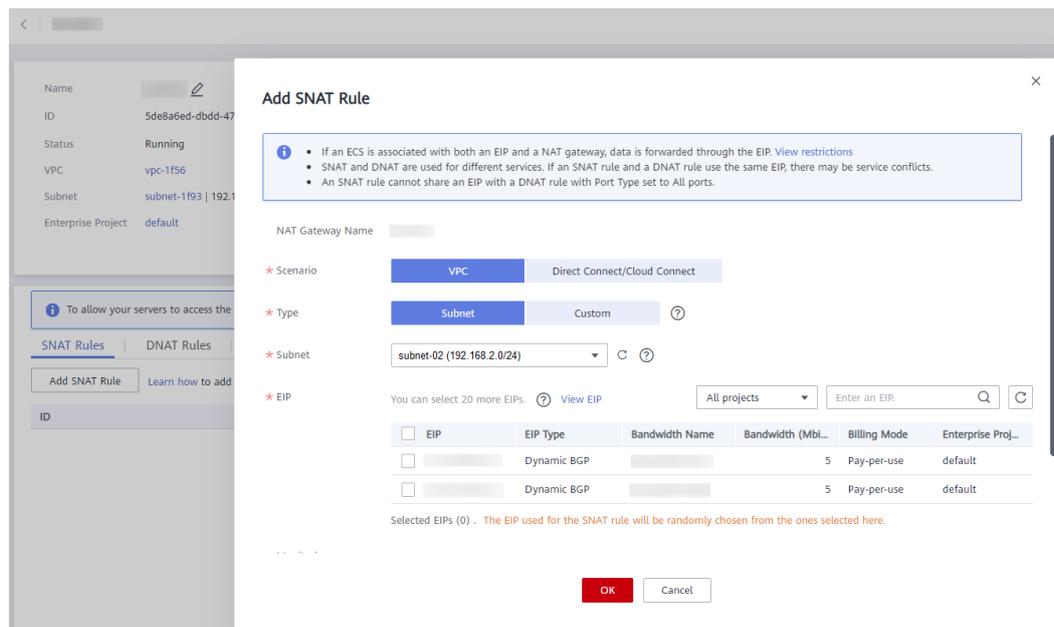
**📖 NOTA**

Las reglas de SNAT entran en vigor por el bloque CIDR. Como los diferentes modelos de red contenedor utilizan diferentes modos de comunicación, la subred debe seleccionarse de acuerdo con las siguientes reglas:

- Red de túnel y red de VPC: Seleccione la subred donde se encuentra el nodo, es decir, la subred seleccionada durante la creación del nodo.

Si hay varios bloques CIDR, puede crear varias reglas de SNAT o personalizar un bloque CIDR siempre que el bloque CIDR contenga la subred de nodo.

**Figura 7-40** Adición de una regla SNAT



Una vez configurada la regla SNAT, las cargas de trabajo pueden acceder a las redes públicas desde contenedor. Las redes públicas se pueden hacer pings desde el contenedor.

----Fin

# 8 Almacenamiento de contenedores

## 8.1 Conceptos básicos del almacenamiento

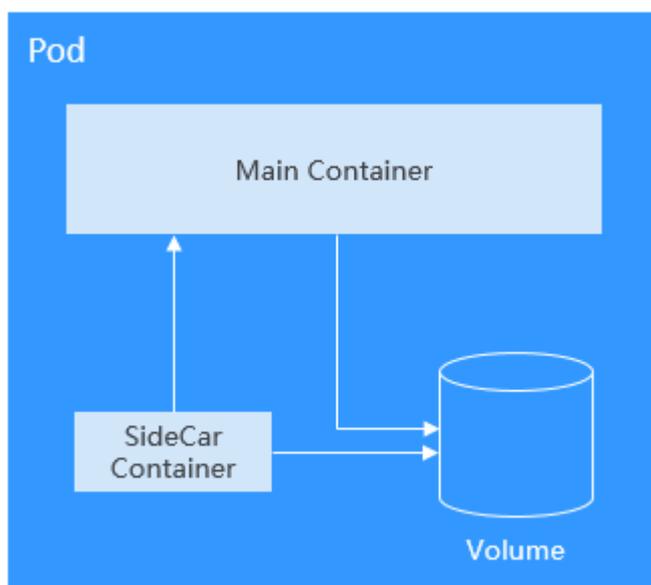
### Volumen

Los archivos en disco de un contenedor son efímeros, lo que presenta los siguientes problemas a las aplicaciones importantes que se ejecutan en el contenedor:

1. Cuando se reconstruye un contenedor, se perderán los archivos del contenedor.
2. Cuando se ejecutan varios contenedores en un pod al mismo tiempo, los archivos deben compartirse entre los contenedores.

La abstracción de volumen de Kubernetes resuelve estos dos problemas. Los volúmenes, como parte de un pod, no se pueden crear de forma independiente y solo se pueden definir en pods. Todos los contenedores de un pod pueden acceder a sus volúmenes, pero los volúmenes deben haber sido montados en cualquier directorio de un contenedor.

La siguiente figura muestra cómo se utiliza un volumen de almacenamiento entre contenedores en un pod.



Los principios básicos para el uso de volúmenes son los siguientes:

- Se pueden montar varios volúmenes en un pod. Sin embargo, no monte demasiados volúmenes en un pod.
- Se pueden montar varios tipos de volúmenes en un pod.
- Cada volumen montado en un pod se puede compartir entre los contenedores en el pod.
- Se recomienda utilizar PVC y PV para montar volúmenes para Kubernetes.

 **NOTA**

El ciclo de vida de un volumen es el mismo que el del pod en el que se monta el volumen. Cuando se elimina el pod, también se elimina el volumen. Sin embargo, los archivos en el volumen pueden sobrevivir al volumen, dependiendo del tipo de volumen.

Kubernetes proporciona varios tipos de volumen. En la siguiente tabla se enumeran los tipos disponibles en CCE.

Clasificación de volumen	Descripción
Almacenamiento local	Volúmenes locales, como hostPath y emptyDir. Para este tipo de volumen, los datos se almacenan en un nodo específico de un clúster y no pueden migrar con aplicaciones. Cuando el nodo se apaga, los datos ya no están disponibles.
Almacenamiento en la nube	Volúmenes en la nube, como SFS y OBS. Para este tipo de volumen, los datos no se almacenan en un nodo de un clúster, sino en un servicio de almacenamiento remoto, y es necesario montar el servicio de almacenamiento en los directorios locales.
Secreto y ConfigMap	Secretos y ConfigMaps son los volúmenes especiales. Almacenan los datos del objeto de los clústeres. Los datos del objeto se montan en los nodos como volúmenes para que las aplicaciones los usen.
PVC	Modo de definición de volumen que abstrae un volumen como un objeto independiente de un pod. La información de almacenamiento definida (asociada) por el objeto es la información de almacenamiento real del volumen y se utiliza para montar cargas de trabajo de Kubernetes.

## PV y PVC

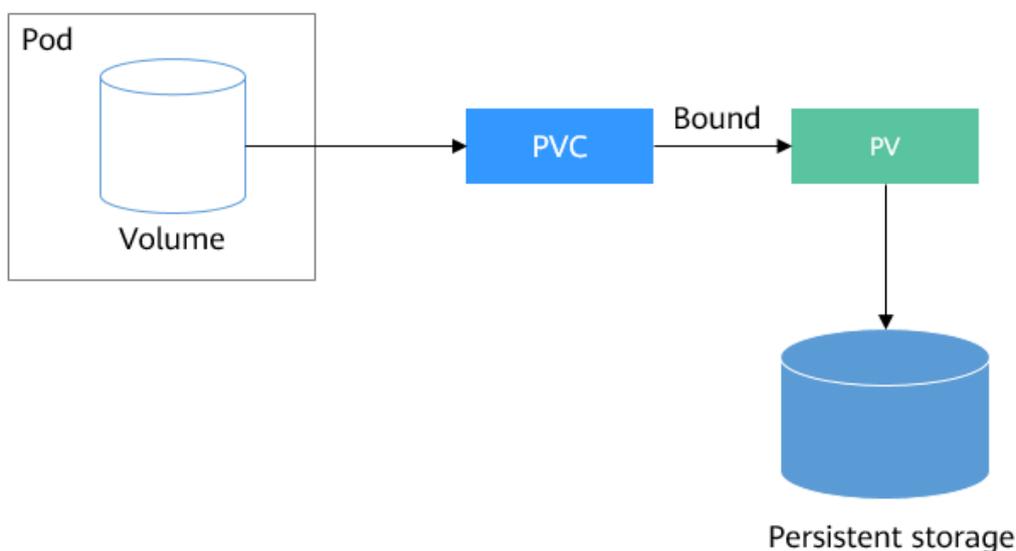
Kubernetes proporciona PersistentVolumes (PV) y PersistentVolumeClaims (PVC) para resumir los detalles de cómo se proporciona el almacenamiento a partir de cómo se consume. Puede solicitar un tamaño específico de almacenamiento cuando sea necesario, al igual que los pods pueden solicitar niveles específicos de recursos (CPU y memoria).

- **PV:** describe un volumen de almacenamiento persistente en un clúster. Un PV es un recurso a nivel de clúster al igual que un nodo. Se aplica a todo el clúster de Kubernetes. Un PV tiene un ciclo de vida independiente de cualquier pod individual que utilice el PV.
- **PVC:** describe una solicitud de almacenamiento por parte de un usuario. Al configurar el almacenamiento para una aplicación, debe reclamar una solicitud de almacenamiento (es decir, PVC). Kubernetes selecciona un PV que mejor se adapte a la solicitud y vincula el

fotovoltaico al PVC. Una unión de PVC a PV es un mapeo uno a uno. Al crear un PVC, debe describir los atributos del almacenamiento persistente solicitado, como el tamaño de almacenamiento y el permiso de lectura/escritura.

Puede vincular los PVC a los PV en un pod para que el pod pueda usar recursos de almacenamiento. La siguiente figura muestra la relación entre los PV y los PVC.

**Figura 8-1** Vinculación de PVC a PV



## Modos de montaje de volúmenes

Puede montar volúmenes de las siguientes maneras:

Utilice PV para describir los recursos de almacenamiento existentes y, a continuación, cree PVC para utilizar los recursos de almacenamiento en los pods. También puede utilizar el modo de creación dinámica. Es decir, especifique el **StorageClass** al crear un PVC y utilice el aprovisionamiento en el StorageClass para crear automáticamente un PV y vincular el PV al PVC.

**Tabla 8-1** Modos de montaje de volúmenes

Modo de montaje	Descripción	Tipo de volumen admitido	Otras restricciones
Creación estática de volumen de almacenamiento (utilizando el almacenamiento existente)	Utilice el almacenamiento existente (como los discos de EVS y los sistemas de archivos SFS) para crear PV y montar los PV en la carga de trabajo con PVC. Kubernetes vincula los PVC a los PV correspondientes para que las cargas de trabajo puedan acceder a los servicios de almacenamiento.	Todos los volúmenes	Ninguna

Modo de montaje	Descripción	Tipo de volumen admitido	Otras restricciones
Creación dinámica de volúmenes de almacenamiento (creación automática de almacenamiento)	Especifique un <a href="#">StorageClass</a> para un PVC. El aprovisionamiento de almacenamiento crea medios de almacenamiento subyacentes según sea necesario para crear automáticamente PV y vincular directamente el PV al PVC.	EVS, OBS, SFS y PV local	Ninguna
Montaje dinámico (VolumeClaim Template)	Logrado mediante el uso del campo <a href="#">volumeClaimTemplates</a> y depende de la capacidad de creación de PV dinámico de StorageClass. En este modo, cada pod está asociado con un PVC y un PV únicos. Después de reprogramar un pod, los datos originales todavía se pueden montar en él basándose en el nombre de PVC.	EVS y PV local	Solo soportado por StatefulSets

## Documentación

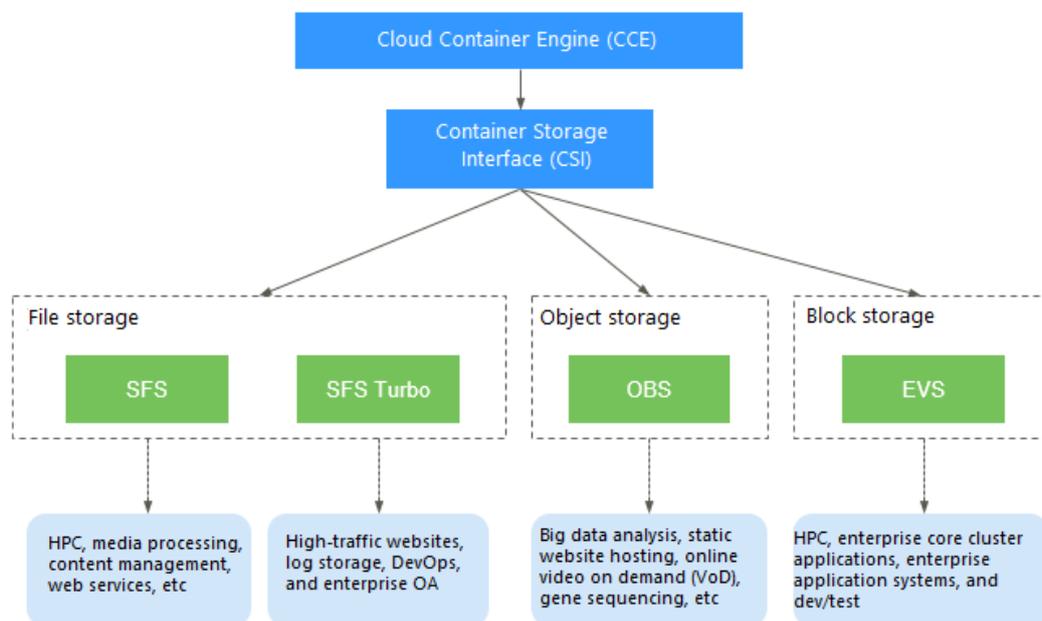
- Para obtener más información sobre el almacenamiento de Kubernetes, consulte [Almacenamiento](#).
- Para obtener más información acerca del almacenamiento contenedor de CCE, vea [Descripción del almacenamiento de contenedores](#).

## 8.2 Descripción del almacenamiento de contenedores

### Servicios en la nube para almacenamiento de contenedores

La función de almacenamiento contenedor de CCE se implementa basada en las API de almacenamiento contenedor de Kubernetes([CSI](#)). CCE integra múltiples tipos de almacenamiento en la nube y cubre diferentes escenarios de aplicación. CCE es totalmente compatible con los servicios de almacenamiento nativos de Kubernetes, como emptyDir, hostPath, secret y ConfigMap.

**Figura 8-2** Tipo de almacenamiento de contenedores



CCE le permite montar volúmenes de almacenamiento en la nube en sus pods. Sus características se describen a continuación.

**Tabla 8-2** Comparación de almacenamiento en la nube

Dimensión	EVS	SFS	OBS	SFS Turbo
Definición	EVS permite ofrecer almacenamiento en bloque ajustable para servidores de nube. Con alta confiabilidad, alto rendimiento y especificaciones ricas, los discos de EVS se pueden utilizar para los sistemas de archivos distribuidos, entornos de desarrollo/prueba, almacenes de datos y aplicaciones informáticas de alto rendimiento (HPC).	Ampliable a petabytes, SFS proporciona almacenamiento de archivos compartidos totalmente alojado, altamente disponible y estable para manejar aplicaciones de uso intensivo de datos y ancho de banda en HPC, procesamiento de medios, uso compartido de archivos, gestión de contenido y servicios web.	OBS es un servicio de almacenamiento de objetos estable, seguro y fácil de usar que le permite almacenar datos de cualquier formato y tamaño de forma económica. Puede usarlo en backup/archivo empresarial, video bajo demanda (VoD), videovigilancia y muchos otros escenarios.	Ampliable a 320 TB, SFS Turbo proporciona un almacenamiento de archivos compartido totalmente alojado, altamente disponible y estable para admitir archivos y aplicaciones pequeños que requieren baja latencia y alta IOPS. Puede utilizar SFS Turbo en sitios web de alto tráfico, almacenamiento de registros, compresión/descompresión, aplicaciones de DevOps, OA empresarial y aplicaciones en contenedores.
Lógica de almacenamiento de datos	Almacena datos binarios y no puede almacenar archivos directamente. Para almacenar archivos, primero debe formatear el sistema de archivos.	Almacena archivos y ordena y muestra datos en la jerarquía de archivos y carpetas.	Almacena objetos. Los archivos almacenados directamente generan automáticamente los metadatos del sistema, que también pueden ser personalizados por los usuarios.	Almacena archivos y ordena y muestra datos en la jerarquía de archivos y carpetas.

Dimensión	EVS	SFS	OBS	SFS Turbo
Modo de acceso	Accesible solo después de ser montado en ECS o BMS e inicializado.	Montado en ECS o BMS usando protocolos de red. Una dirección de red debe especificarse o asignarse a un directorio local para tener acceso.	Accesible con Internet o Direct Connect (DC). Debe especificar la dirección del bucket y utilizar protocolos de transmisión como HTTP y HTTPS.	Admite el protocolo Network File System (NFS) (solo NFSv3). Puede integrar sin problemas las aplicaciones y herramientas existentes con SFS Turbo.
Aprovisionamiento estático	Se admite. Para obtener más información, véase <a href="#">Uso de un disco de EVS existente a través de un PV estático</a> .	Se admite. Para obtener más información, véase <a href="#">Uso de un sistema de archivos de SFS existente con un PV estático</a> .	Se admite. Para obtener más información, véase <a href="#">Uso de un bucket de OBS existente con un PV estático</a> .	Se admite. Para obtener más información, véase <a href="#">Uso de un sistema de archivos de SFS Turbo existente con un PV estático</a> .
Aprovisionamiento dinámico	Se admite. Para obtener más información, véase <a href="#">Uso de un disco de EVS con un PV dinámico</a> .	Se admite. Para obtener más información, véase <a href="#">Uso de un sistema de archivos SFS a través de un PV dinámico</a> .	Se admite. Para obtener más información, véase <a href="#">Uso de un bucket de OBS con un PV dinámico</a> .	No se admite
Funciones	Almacenamiento no compartido. Cada volumen se puede montar en un solo nodo.	Almacenamiento compartido de alto rendimiento y capacidad	Sistema de archivos compartido en modo usuario	Almacenamiento compartido con alto rendimiento y ancho de banda

Dimensión	EVS	SFS	OBS	SFS Turbo
Uso	HPC, aplicaciones de clúster de núcleo empresarial, sistemas de aplicaciones empresariales y desarrollo/prueba <b>NOTA</b> Las aplicaciones de HPC requieren almacenamiento de alta velocidad y alta IOPS, como el diseño industrial y la exploración de energía.	HPC, procesamiento de medios, gestión de contenido, servicios web, big data y aplicaciones de análisis <b>NOTA</b> Las aplicaciones de HPC requieren un alto ancho de banda y almacenamiento de archivos compartidos, como la secuenciación de genes y el renderizado de imágenes.	Análisis de big data, alojamiento de sitios web estático, video on demand en línea (VoD), secuenciación de genes, videovigilancia inteligente, copia de respaldo y archivado, y cajas en la nube empresarial (discos web)	Sitios web de alto tráfico, almacenamiento de registros, DevOps y OA empresarial
Capacidad	TB	SFS 1.0: PB	EB	Uso general: TB
Latencia	1–2 ms	SFS 1.0: 3–20 ms	10 ms	Uso general: 1–5 ms
IOPS/TPS	33,000 para un solo disco	SFS 1.0: 2,000	Decenas de millones	Uso general: hasta 100,000
Ancho de banda	MB/s	SFS 1.0: GB/s	TB/s	Uso general: hasta GB/s

## CSI

Container Storage Interface (CSI) es un estándar para las interfaces de almacenamiento contenedor y una solución de implementación de complementos de almacenamiento recomendada por la comunidad Kubernetes. [everest](#) es un complemento de almacenamiento desarrollado por Huawei basado en CSI. Proporciona diferentes tipos de almacenamiento persistente para contenedores.

## Modos de acceso al volumen

Los PV se pueden montar en el sistema host solo en el modo soportado por los recursos de almacenamiento subyacentes. Por ejemplo, un sistema de almacenamiento de archivos puede leerse y escribirse por múltiples nodos, pero un disco de EVS puede leerse y escribirse por un solo nodo.

- **ReadWriteOnce:** Un volumen puede ser montado como lectura-escritura por un solo nodo.

- **ReadWriteMany:** Un volumen puede ser montado como lectura-escritura por muchos nodos.

**Tabla 8-3** Modos de acceso compatibles con los volúmenes de almacenamiento

Tipo de almacenamiento	ReadWriteOnce	ReadWriteMany
EVS	√	×
SFS	×	√
OBS	×	√
SFS Turbo	×	√
Local PV	√	×

## Política de reclamo de PV

Una política de reclamo de PV se utiliza para eliminar o reclamar volúmenes subyacentes cuando se elimina un PVC. El valor puede ser **Delete** o **Retain**.

- **Delete:** Al eliminar un PVC se quitará el PV de Kubernetes, por lo que los activos de almacenamiento asociados de la infraestructura externa.

### **NOTA**

Los recursos anuales/mensuales no se pueden eliminar mediante la política de recuperación **Delete**.

- **Retain:** Cuando se elimina un PVC, el PV y los recursos de almacenamiento subyacentes no se eliminan. En su lugar, debe eliminar manualmente estos recursos. Después de eso, los recursos de PV están en el estado **Released** y no se pueden vincular directamente al PVC.

Puede eliminar y recuperar volúmenes manualmente mediante el procedimiento siguiente:

- Elimine el PV.
- Borre los datos de los recursos de almacenamiento subyacentes asociados según sea necesario.
- Elimine los recursos de almacenamiento subyacentes asociados.

Para reutilizar los recursos de almacenamiento subyacentes, cree un PV.

Everest también le permite eliminar un PVC sin eliminar los recursos de almacenamiento subyacentes. Esta función solo se puede lograr usando un archivo YAML: Establezca la política de recuperación de PV a **Delete** y agregue **everest.io/reclaim-policy: retain-volume-only** a **annotations**. De esta manera, cuando se elimina el PVC, se elimina el PV, pero se conservan los recursos de almacenamiento subyacentes.

El siguiente archivo YAML toma EVS como ejemplo.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: test
  namespace: default
```

```

annotations:
  volume.beta.kubernetes.io/storage-provisioner: everest-csi-provisioner
  everest.io/disk-volume-type: SAS
labels:
  failure-domain.beta.kubernetes.io/region: <your_region> # Region of the
node where the application is to be deployed.
  failure-domain.beta.kubernetes.io/zone: <your_zone> # AZ of the node
where the application is to be deployed.
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
  storageClassName: csi-disk
  volumeName: pv-evs-test
---
apiVersion: v1
kind: PersistentVolume
metadata:
  annotations:
    pv.kubernetes.io/provisioned-by: everest-csi-provisioner
    everest.io/reclaim-policy: retain-volume-only
  name: pv-evs-test
  labels:
    failure-domain.beta.kubernetes.io/region: <your_region> # Region of the
node where the application is to be deployed.
    failure-domain.beta.kubernetes.io/zone: <your_zone> # AZ of the node
where the application is to be deployed.
spec:
  accessModes:
    - ReadWriteOnce
  capacity:
    storage: 10Gi
  csi:
    driver: disk.csi.everest.io
    fsType: ext4
    volumeHandle: 2af98016-6082-4ad6-bedc-1a9c673aef20
    volumeAttributes:
      storage.kubernetes.io/csiProvisionerIdentity: everest-csi-provisioner
      everest.io/disk-mode: SCSI
      everest.io/disk-volume-type: SAS
    persistentVolumeReclaimPolicy: Delete
  storageClassName: csi-disk
    
```

## Soporte de proyectos empresariales

### NOTA

Para utilizar esta función, el complemento everest debe actualizarse a 1.2.33 o posterior.

- Creación automática de almacenamiento:

CCE permite especificar un proyecto de empresa al crear discos de EVS y PVC de OBS. Los recursos de almacenamiento creados (discos de EVS y OBS) pertenecen al proyecto de empresa especificado. **El proyecto de empresa puede ser el proyecto de empresa al que pertenece el cluster o el proyecto de empresa por defecto.**

Si no se especifica ningún proyecto de empresa, el proyecto de empresa especificado en StorageClass se utilizará de forma predeterminada para crear recursos de almacenamiento.

- Para un StorageClass personalizado, puede especificar un proyecto de empresa de StorageClass. Para obtener más información, véase [Especificación de un proyecto de empresa para clases de almacenamiento](#). Si StorageClass no especifica ningún proyecto de empresa, se utilizará el proyecto de empresa predeterminado.

- Para las clases de almacenamiento `csi-disk` y `csi-obs` proporcionadas por CCE, los recursos de almacenamiento creados pertenecen al proyecto de empresa predeterminado.
- Utilice el almacenamiento existente:  
Cuando cree un PVC mediante un PV, asegúrese de que los **everest.io/enterprise-project-id** especificados en el PVC y el PV son los mismos porque se ha especificado un proyecto de empresa durante la creación de recursos de almacenamiento. De lo contrario, el PVC y el PV no se pueden unir.

## Descripción de la versión del complemento

Para utilizar el complemento CSI (**everest**), la versión de Kubernetes debe ser 1.15 o posterior.

El complemento Flexvolume (**storage-driver**) se instala de forma predeterminada cuando se crea un clúster de v1.13 o anterior. Si el clúster se actualiza de v1.13 a v1.15, el **storage-driver** se sustituye por `everest` (v1.1.6 o posterior) para el almacenamiento de contenedor. La adquisición no afecta a las funciones de almacenamiento originales.

## Diferencias entre los complementos de CSI y de FlexVolume

Tabla 8-4 CSI y FlexVolume

Solución de Kubernetes	Complementos de CCE	Funciones	Recomendación
CSI	everest	<p>CSI se desarrolló como un estándar para exponer los sistemas de almacenamiento de archivos y bloques arbitrarios a cargas de trabajo en contenedores. Con CSI, los proveedores de almacenamiento de terceros pueden desplegar complementos que expongan nuevos sistemas de almacenamiento en Kubernetes sin tener que tocar el código principal de Kubernetes. En CCE, el complemento everest se instala de forma predeterminada en clústeres de Kubernetes v1.15 y posteriores para conectarse a servicios de almacenamiento (EVS, OBS, SFS y SFS Turbo).</p> <p>El complemento everest consta de dos partes:</p> <ul style="list-style-type: none"> <li>● <b>everest-csi-controller</b> para creación de volúmenes de almacenamiento, eliminación, ampliación de la capacidad e instantáneas de disco en la nube</li> <li>● <b>everest-csi-driver</b> para montar, desmontar y formatear volúmenes de almacenamiento en nodos</li> </ul> <p>Para más detalles, consulte <a href="#">everest</a>.</p>	<p>El complemento <b>everest</b> se instala de forma predeterminada en clústeres de <b>v1.15 y posteriores</b>. CCE reflejará la comunidad de Kubernetes al proporcionar soporte continuo para las capacidades de CSI actualizadas.</p>

Solución de Kubernetes	Complementos de CCE	Funciones	Recomendación
FlexVolume	storage-driver	FlexVolume es una interfaz de complemento fuera de árbol que existe en Kubernetes desde la versión 1.2 (antes de CSI). CCE proporcionó volúmenes de FlexVolume a través del complemento del controlador de almacenamiento instalado en clústeres de Kubernetes v1.13 y las versiones anteriores. Este complemento conecta clústeres a servicios de almacenamiento (EVS, OBS, SFS y SFS Turbo).  Para obtener más información, consulte <a href="#">storage-driver</a> .	Para los clústeres creados de <b>v1.13 o anteriores</b> , el complemento de FlexVolume instalado (complemento <a href="#">storage-driver</a> de CCE) todavía se puede utilizar. CCE deja de proporcionar soporte de actualización para este complemento, y se recomienda que <a href="#">actualice estos clústeres</a> .

 **NOTA**

- Un clúster solo puede utilizar un tipo de complementos de almacenamiento.
- El complemento de FlexVolume no puede ser reemplazado por el complemento de CSI en clusters de v1.13 o anteriores. Solo puede actualizar estos clústeres. Para obtener más información, véase [Actualización de clúster](#).

## Comprobación de complementos de almacenamiento

**Paso 1** Inicie sesión en la consola de CCE.

**Paso 2** Haga clic en el nombre del clúster para acceder a la consola del clúster. Elija **Add-ons** en el panel de navegación.

**Paso 3** Haga clic en la ficha **Add-on Instance**.

**Paso 4** Seleccione un clúster en la esquina superior derecha. Se muestra el complemento de almacenamiento predeterminado instalado durante la creación del clúster.

----Fin

## Documentación

- [Conceptos básicos del almacenamiento](#)
- [Elastic Volume Service \(EVS\)](#)
- [Scalable File Service \(SFS\)](#)
- [Sistemas de archivos SFS Turbo](#)
- [Object Storage Service \(OBS\)](#)

## 8.3 Elastic Volume Service (EVS)

### 8.3.1 Descripción general

Para lograr el almacenamiento persistente, CCE le permite montar los volúmenes de almacenamiento creados a partir de los discos de Elastic Volume Service (EVS) en una ruta de un contenedor. Cuando el contenedor se migra dentro de una AZ, los volúmenes de EVS montados también se migran. Mediante el uso de volúmenes de EVS, puede montar el directorio de archivos remoto de un sistema de almacenamiento en un contenedor para que los datos del volumen de datos se conserven permanentemente. Incluso si se elimina el contenedor, los datos en el volumen de datos todavía se almacenan en el sistema de almacenamiento.

### Especificaciones de rendimiento del disco de EVS

Las métricas de rendimiento de EVS incluyen:

- IOPS: número de operaciones de lectura/escritura realizadas por un disco de EVS por segundo
- Rendimiento: Cantidad de datos leídos y escritos en un disco EVS por segundo
- Latencia de E/S de lectura/escritura: Intervalo mínimo entre dos operaciones consecutivas de lectura/escritura en un disco EVS

**Tabla 8-5** Especificaciones de rendimiento del disco de EVS

Parámetro	SSD con capacidad extrema	SSD de uso general	Capacidad ultraalta de E/S	Capacidad alta de E/S
Máx. capacidad (GiB)	<ul style="list-style-type: none"> <li>● Disco del sistema: 1,024</li> <li>● Disco de datos: 32,768</li> </ul>	<ul style="list-style-type: none"> <li>● Disco del sistema: 1,024</li> <li>● Disco de datos: 32,768</li> </ul>	<ul style="list-style-type: none"> <li>● Disco del sistema: 1,024</li> <li>● Disco de datos: 32,768</li> </ul>	<ul style="list-style-type: none"> <li>● Disco del sistema: 1,024</li> <li>● Disco de datos: 32,768</li> </ul>
Máx. IOPS	128,000	20,000	50,000	5,000
Máx. rendimiento (MiB/s)	1,000	250	350	150
Límite de IOPS de ráfaga	64,000	8,000	16,000	5,000
IOPS del disco	Mín. (128,000, 1,800 + 50 x capacidad)	Mín. (20,000, 1,800 + 12 x capacidad)	Mín. (50,000, 1,800 + 50 x capacidad)	Mín. (5,000, 1,800 + 8 x capacidad)

Parámetro	SSD con capacidad extrema	SSD de uso general	Capacidad ultraalta de E/S	Capacidad alta de E/S
Rendimiento del disco (MiB/s)	Mín. (1,000, 120 + 0.5 × capacidad)	Mín. (250, 100 + 0.5 × capacidad)	Mín. (350, 120 + 0.5 × capacidad)	Mín. (150, 100 + 0.15 × capacidad)
Latencia de acceso de una sola cola (ms)	Submilisegundo	1	1	1–3
Nombre de la API	ESSD	GPSSD	SSD	SAS

Para obtener más información sobre el rendimiento del disco de EVS, consulte [Tipos y rendimiento de disco](#).

## Escenarios de aplicación

Los discos de EVS se pueden montar en los siguientes modos basados en escenarios de aplicación:

- **Uso de un disco de EVS existente a través de un PV estático:** modo de creación estática, donde se utiliza un disco de EVS existente para crear un PV y luego montar el almacenamiento en la carga de trabajo con un PVC. Este modo es aplicable a escenarios en los que el almacenamiento subyacente está disponible o se factura anualmente/mensualmente.
- **Uso de un disco de EVS con un PV dinámico:** modo de creación dinámica, donde no es necesario crear los volúmenes de EVS por adelantado. En su lugar, especifique un StorageClass durante la creación de PVC y un disco de EVS y un PV se crearán automáticamente. Este modo es aplicable a escenarios en los que no hay almacenamiento subyacente disponible.
- **Montaje dinámico de un disco de EVS en un StatefulSet:** Solo StatefulSets admite este modo. Cada pod está asociado con un PVC y PV únicos. Después de reprogramar un pod, los datos originales todavía se pueden montar en él basándose en el nombre de PVC. Este modo es aplicable a StatefulSets con múltiples pods.

## Restricciones

- Los discos de EVS no se pueden montar entre las AZ y no se pueden usar por varias cargas de trabajo, varios pods de la misma carga de trabajo o varias tareas.
- La función de intercambio de datos de un disco compartido no se admite entre los nodos de un clúster de CCE. Si el mismo disco de EVS está montado en varios nodos, pueden producirse conflictos de lectura y escritura y conflictos de caché de datos. Por lo tanto, se recomienda crear solo un pod al crear una Deployment que utilice discos de EVS.
- Para los clústeres anteriores a v1.19.10, si se utiliza una política de HPA para escalar una carga de trabajo con volúmenes de EVS montados, los pods existentes no se pueden leer ni escribir cuando se programa un nuevo pod en otro nodo.

Para los clústeres de v1.19.10 y las versiones posteriores, si se utiliza una política de HPA para escalar una carga de trabajo con un volumen de EVS montado, no se puede iniciar un nuevo pod porque no se pueden conectar los discos de EVS.

- Los discos de EVS que tienen particiones o utilizan sistemas de archivos no ext4 no son compatibles.
- El almacenamiento de contenedores en clústeres de CCE de Kubernetes 1.13 o posterior admite la encriptación. Antes de utilizar el almacenamiento contenedor, compruebe si la región donde se encuentra el disco EVS admite la encriptación de disco.

## Facturación

- Para montar volúmenes de almacenamiento del tipo de EVS, el modo de facturación de los discos de EVS **creó automáticamente** especificando el StorageClass es de pago por uso de forma predeterminada y no se puede cambiar a anual/mensual. Si desea utilizar un disco de EVS con facturación anual/mensual, **utilice uno existente**.
- Para obtener más información sobre los precios de los discos EVS, consulte **Facturación para discos**.

## 8.3.2 Uso de un disco de EVS existente a través de un PV estático

CCE le permite crear un PV utilizando un disco de EVS existente. Una vez creado el PV, puede crear un PVC y vincularlo al PV. Este modo es aplicable a escenarios en los que el almacenamiento subyacente está disponible o se factura anualmente/mensualmente.

### Requisitos previos

- Ha creado un clúster e instalado el complemento de CSI (**everest**) en el clúster.
- Ha creado un disco de EVS que cumple con los requisitos descritos en **Restricciones**.
- Si desea crear un clúster mediante comandos, utilice kubectl para conectarse al clúster. Para obtener más información, véase **Conexión a un clúster con kubectl**.

### Restricciones

- El disco de EVS existente no puede ser un disco del sistema, un disco de DSS o un disco compartido.
- El disco de EVS debe ser uno de los tipos admitidos en la región actual (E/S común, E/S alta, SSD de uso general, SSD extremo y E/S ultraalta) y el tipo de dispositivo de disco de EVS debe ser SCSI.
- El disco de EVS debe estar disponible y no ser utilizado por otros recursos.
- La AZ del disco de EVS debe ser la misma que la del nodo del clúster. De lo contrario, el disco de EVS no se puede montar y el pod no se puede iniciar.
- Si el disco de EVS está cifrado, la clave debe estar disponible.
- Solo se admiten los discos de EVS del proyecto de empresa al que pertenece el clúster y el proyecto de empresa predeterminado.

### (Consola) Uso de un disco de EVS existente

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** Cree un PV y PVC estáticamente.

1. Elija **Storage** en el panel de navegación y haga clic en la ficha **PersistentVolumeClaims (PVCs)**. Haga clic en **Create PVC** en la esquina superior derecha. En el cuadro de diálogo que se muestra, configure los parámetros de PVC.

Parámetro	Descripción
PVC Type	En este ejemplo, seleccione <b>EVS</b> .
PVC Name	Escriba el nombre de PVC, que debe ser único en el mismo espacio de nombres.
Creation Method	<ul style="list-style-type: none"> <li>– Si el almacenamiento subyacente está disponible, puede crear un volumen de almacenamiento o utilizar un volumen de almacenamiento existente para crear estáticamente un PVC en función de si se ha creado un PV.</li> <li>– Si no hay ningún almacenamiento subyacente disponible, puede seleccionar <b>Dynamically provision</b>. Para obtener más información, véase <a href="#">Uso de un disco de EVS con un PV dinámico</a>.</li> </ul> <p>En este ejemplo, seleccione <b>Create new</b> para crear un PV y un PVC al mismo tiempo en la consola.</p>
PV <sup>a</sup>	<p>Seleccione un PV existente en el clúster. Es necesario crear un PV por adelantado. Para obtener más información, consulte "Creación de un volumen de almacenamiento (PV)" de <a href="#">Operaciones relacionadas</a>.</p> <p>En este ejemplo, no es necesario establecer este parámetro.</p>
EVS <sup>b</sup>	Haga clic en <b>Select EVS</b> . En la página mostrada, seleccione el disco de EVS que cumpla con sus requisitos y haga clic en <b>OK</b> .
PV Name <sup>b</sup>	Introduzca el nombre de PV, que debe ser único en el mismo clúster.
Access Mode <sup>b</sup>	Los discos de EVS solo admiten <b>ReadWriteOnce</b> , lo que indica que un volumen de almacenamiento se puede montar en un nodo en modo de lectura/escritura. Para obtener más información, véase <a href="#">Modos de acceso al volumen</a> .
Reclaim Policy <sup>b</sup>	Puede seleccionar <b>Delete</b> o <b>Retain</b> para especificar la política de recuperación del almacenamiento subyacente cuando se elimina el PVC. Para obtener más información, véase <a href="#">Política de reclamo de PV</a> .

 **NOTA**

a: El parámetro está disponible cuando **Creation Method** se establece en **Use existing**.

b: El parámetro está disponible cuando **Creation Method** se establece en **Create new**.

2. Haga clic en **Create** para crear un PVC y un PV.

Puede elegir **Storage** en el panel de navegación y ver el PVC y PV creados en las páginas de fichas **PersistentVolumeClaims (PVCs)** y **PersistentVolumes (PVs)**.

**Paso 3** Cree una aplicación.

1. En el panel de navegación de la izquierda, haga clic en **Workloads**. En el panel derecho, haga clic en la ficha **StatefulSets**.
2. Haga clic en **Create Workload** en la esquina superior derecha. En la página mostrada, haga clic en **Data Storage** en el área **Container Settings** y haga clic en **Add Volume** para seleccionar **PVC**.

Montar y utilizar volúmenes de almacenamiento, como se muestra en [Tabla 8-6](#). Para obtener más información sobre otros parámetros, consulte [Workloads](#).

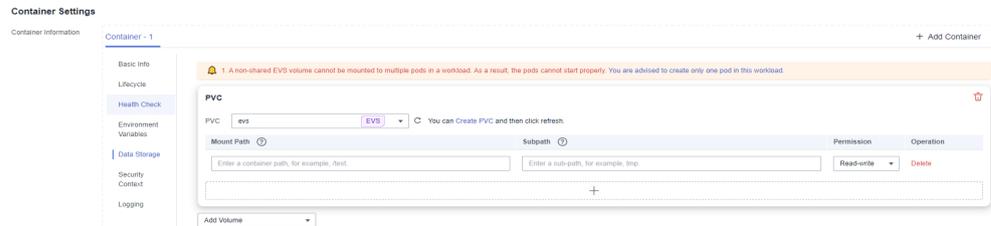
**Tabla 8-6** Montaje de un volumen de almacenamiento

Parámetro	Descripción
PVC	<p>Seleccione un volumen de EVS existente.</p> <p>Un volumen de EVS no se puede montar repetidamente en varias cargas de trabajo.</p>
Add Container Path	<p>Configure los parámetros siguientes:</p> <ol style="list-style-type: none"> <li>1. <b>Mount Path</b>: Introduzca una ruta de acceso del contenedor. Por ejemplo, <b>/tmp</b>. Este parámetro indica la ruta de contenedor en la que se montará un volumen de datos. No monte el volumen en un directorio del sistema como <b>/</b> o <b>/var/run</b>; esta acción puede causar errores de contenedor. Se recomienda montar el volumen en un directorio vacío. Si el directorio no está vacío, asegúrese de que no hay archivos que afecten al inicio de contenedor. De lo contrario, se reemplazarán los archivos, lo que provocará errores de inicio de contenedor o errores de creación de carga de trabajo.                     <p><b>AVISO</b></p> <p>Cuando el contenedor está montado en un directorio de alto riesgo, se recomienda utilizar una cuenta con los permisos mínimos para iniciar el contenedor; de lo contrario, los archivos de alto riesgo en la máquina host podrían estar dañados.</p> </li> <li>2. <b>Subpath</b>: Introduzca una ruta secundaria, por ejemplo, <b>tmp</b>. Se utiliza una ruta secundaria para montar un volumen local de modo que se utilice el mismo volumen de datos en un solo pod. Si este parámetro se deja en blanco, la ruta raíz se utiliza de forma predeterminada.</li> <li>3. <b>Permission</b> <ul style="list-style-type: none"> <li>■ <b>Read-only</b>: Solo puede leer los datos en los volúmenes montados.</li> <li>■ <b>Read/Write</b>: Puede modificar los volúmenes de datos montados en la ruta de acceso. Los datos recién escritos no se migran si se migra el contenedor, lo que puede causar la pérdida de datos.</li> </ul> </li> </ol> <p>Puede hacer clic en  para agregar varias rutas y subrutas.</p>

En este ejemplo, el disco se monta en la trayectoria **/data** del contenedor. Los datos de contenedor generados en esta ruta se almacenan en el disco de EVS.

**NOTA**

Un disco ReadWriteOnce de EVS no compartido no se puede montar en varios pods de una carga de trabajo. Como resultado, los pods no pueden arrancar correctamente. Asegúrese de que el número de pods de carga de trabajo es 1 al montar un disco de EVS.



3. Después de completar la configuración, haga clic en **Create**.

Una vez creada la carga de trabajo, puede probar [Verificación de la persistencia de datos](#).

----Fin

## (kubectl) Uso de un disco de EVS existente

**Paso 1** Utilice kubectl para conectarse al clúster.

**Paso 2** Crear un PV.

1. Cree el archivo **pv-evs.yaml**.

```
apiVersion: v1
kind: PersistentVolume
metadata:
  annotations:
    pv.kubernetes.io/provisioned-by: everest-csi-provisioner
    everest.io/reclaim-policy: retain-volume-only # (Optional) The PV
is deleted while the underlying volume is retained.
  name: pv-evs # PV name.
  labels:
    failure-domain.beta.kubernetes.io/region: <your_region> # Region of the
node where the application is to be deployed.
    failure-domain.beta.kubernetes.io/zone: <your_zone> # AZ of the
node where the application is to be deployed.
spec:
  accessModes:
    - ReadWriteOnce # Access mode. The value is fixed to ReadWriteOnce
for EVS disks.
  capacity:
    storage: 10Gi # EVS disk capacity, in the unit of Gi. The value
ranges from 1 to 32768.
  csi:
    driver: disk.csi.everest.io # Dependent storage driver for the
mounting.
    fsType: ext4
    volumeHandle: <your_volume_id> # Volume ID of the EVS disk.
    volumeAttributes:
      everest.io/disk-mode: SCSI # Device type of the EVS disk.
Only SCSI is supported.
      everest.io/disk-volume-type: SAS # EVS disk type.
      storage.kubernetes.io/csiProvisionerIdentity: everest-csi-provisioner
      everest.io/crypt-key-id: <your_key_id> # (Optional) Encryption key
ID. Mandatory for an encrypted disk.
      everest.io/enterprise-project-id: <your_project_id> # (Optional)
Enterprise project ID. If an enterprise project is specified, you need to use
```

```

the same enterprise project when creating a PVC. Otherwise, the PVC cannot be
bound to a PV.
  persistentVolumeReclaimPolicy: Delete      # Reclaim policy.
  storageClassName: csi-disk                # Storage class name. The value
must be csi-disk for EVS disks.
    
```

**Tabla 8-7** Parámetros clave

Parámetro	Obligatorio	Descripción
everest.io/reclaim-policy: retain-volume-only	No	Opcional. Actualmente, solo se admite <b>retain-volume-only</b> . Este campo solo es válido cuando la versión más reciente es 1.2.9 o posterior y la política de recuperación es <b>Delete</b> . Si la política de reclamación es de <b>Delete</b> y el valor actual es de <b>retain-volume-only</b> el PV asociado se elimina mientras se conserva el volumen de almacenamiento subyacente cuando se elimina un PVC.
failure-domain.beta.kubernetes.io/region	Sí	Región donde se encuentra el clúster. Para obtener más información sobre el valor de <b>region</b> , consulte <a href="#">Regiones y puntos de conexión</a> .
failure-domain.beta.kubernetes.io/zone	Sí	AZ donde se crea el volumen de EVS. Debe ser la misma que la AZ prevista para la carga de trabajo. Para obtener más información sobre el valor de <b>zone</b> , consulte <a href="#">Regiones y puntos de conexión</a> .
volumeHandle	Sí	ID de volumen del disco de EVS. Para obtener el ID de volumen, inicie sesión en <b>Cloud Server Console</b> . En el panel de navegación, elija <b>Elastic Volume Service &gt; Disks</b> . Haga clic en el nombre del disco de EVS de destino para ir a su página de detalles. En la página de ficha <b>Summary</b> , haga clic en el botón de copiar después de <b>ID</b> .
everest.io/disk-volume-type	Sí	Tipo del disco de EVS. Todas las letras están en mayúsculas. <ul style="list-style-type: none"> <li>– <b>SAS</b>: E/S con capacidad alta</li> <li>– <b>SSD</b>: E/S con capacidad ultraalta</li> <li>– <b>GPSSD</b>: SSD de uso general</li> <li>– <b>ESSD</b>: SSD extremo</li> </ul>

Parámetro	Obligatorio	Descripción
everest.io/crypt-key-id	No	<p>Obligatorio cuando el disco de EVS está cifrado. Introduzca el ID de clave de encriptación seleccionado durante la creación del disco de EVS.</p> <p>Para obtener el ID de clave de encriptación, inicie sesión en <b>Cloud Server Console</b>. En el panel de navegación, elija <b>Elastic Volume Service &gt; Disks</b>. Haga clic en el nombre del disco de EVS de destino para ir a su página de detalles. En la página de ficha <b>Summary</b>, copie el valor de <b>KMS Key ID</b> en el área <b>Configuration Information</b>.</p>
everest.io/enterprise-project-id	No	<p>Opcional.</p> <p>ID de proyecto de empresa del disco de EVS. Si se especifica un proyecto de empresa, debe especificar el mismo proyecto de empresa al crear un PVC. De lo contrario, el PVC no puede estar unido a un PV.</p> <p>Para obtener el ID del proyecto de empresa, inicie sesión en <b>Cloud Server Console</b>. En el panel de navegación, elija <b>Elastic Volume Service &gt; Disks</b>. Haga clic en el nombre del disco de EVS de destino para ir a su página de detalles. En la página de ficha <b>Summary</b>, haga clic en el proyecto de empresa de <b>Management Information</b> para acceder a la consola del proyecto de empresa. Copie el ID correspondiente para obtener el ID del proyecto de empresa al que pertenece el disco de EVS.</p>

Parámetro	Obligatorio	Descripción
persistentVolumeReclaimPolicy	Sí	<p>Se admite una política de recuperación cuando la versión del clúster es o posterior a 1.19.10 y la versión everest es o posterior a 1.2.9.</p> <p>Las políticas de recuperación <b>Delete</b> y <b>Retain</b> son compatibles. Para obtener más información, véase <a href="#">Política de reclamo de PV</a>. Si se requiere una alta seguridad de los datos, se recomienda seleccionar <b>Retain</b> para evitar que los datos se eliminen por error.</p> <p><b>Delete:</b></p> <ul style="list-style-type: none"> <li>Si no se especifica <b>everest.io/reclaim-policy</b>, tanto el PV como el disco de EVS se eliminan cuando se elimina un PVC.</li> <li>Si <b>everest.io/reclaim-policy</b> se establece en <b>retain-volume-only set</b> cuando se elimina un PVC, se elimina el PV pero se conservan los recursos de EVS.</li> </ul> <p><b>Retain:</b> Cuando se elimina un PVC, el PV y los recursos de almacenamiento subyacentes no se eliminan. En su lugar, debe eliminar manualmente estos recursos. Después de eso, el PV está en el estado <b>Released</b> y no puede estar ligado al PVC de nuevo.</p>
storageClassName	Sí	El nombre de clase de almacenamiento para los discos de EVS es <b>csi-disk</b> .

- Ejecute el siguiente comando para crear un PV:

```
kubectl apply -f pv-evs.yaml
```

### Paso 3 Cree un PVC.

- Cree el archivo **pvc-evs.yaml**.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-evs
  namespace: default
  annotations:
    everest.io/disk-volume-type: SAS # EVS disk type.
    everest.io/crypt-key-id: <your_key_id> # (Optional) Encryption key ID.
    everest.io/enterprise-project-id: <your_project_id> # (Optional)
Mandatory for an encrypted disk.
Enterprise project ID. If an enterprise project is specified, you need to use
the same enterprise project when creating a PVC. Otherwise, the PVC cannot be
bound to a PV.
  labels:
    failure-domain.beta.kubernetes.io/region: <your_region> # Region of the
node where the application is to be deployed.
    failure-domain.beta.kubernetes.io/zone: <your_zone> # AZ of the
node where the application is to be deployed.
spec:
  accessModes:
```

```
- ReadWriteOnce # The value must be ReadWriteOnce for EVS
disks.
resources:
  requests:
    storage: 10Gi # EVS disk capacity, ranging from 1 to 32768.
The value must be the same as the storage size of the existing PV.
    storageClassName: csi-disk # Storage class type for EVS disks.
    volumeName: pv-evs # PV name.
```

**Tabla 8-8** Parámetros clave

Parámetro	Obligatorio	Descripción
failure-domain.beta.kubernetes.io/region	Sí	Región donde se encuentra el clúster. Para obtener más información sobre el valor de <b>region</b> , consulte <a href="#">Regiones y puntos de conexión</a> .
failure-domain.beta.kubernetes.io/zone	Sí	AZ donde se crea el volumen de EVS. Debe ser la misma que la AZ prevista para la carga de trabajo. Para obtener más información sobre el valor de <b>zone</b> , consulte <a href="#">Regiones y puntos de conexión</a> .
storage	Sí	Capacidad solicitada en el PVC, en Gi. El valor debe ser el mismo que el tamaño de almacenamiento del PV existente.
volumeName	Sí	Nombre de PV, que debe ser el mismo que el nombre de PV en <b>1</b> .
storageClassName	Sí	Nombre de la clase de almacenamiento, que debe ser el mismo que la clase de almacenamiento del PV en <b>1</b> . El nombre de clase de almacenamiento de los volúmenes de EVS es <b>csi-disk</b> .

2. Ejecute el siguiente comando para crear un PVC:

```
kubectl apply -f pvc-evs.yaml
```

**Paso 4** Cree una aplicación.

1. Cree un archivo denominado **web-evs.yaml**. En este ejemplo, el volumen de EVS se monta en la ruta **/data**.

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: web-evs
  namespace: default
spec:
  replicas: 1 # The number of workload replicas that use the EVS
volume must be 1.
  selector:
    matchLabels:
      app: web-evs
  serviceName: web-evs # Headless Service name.
  template:
    metadata:
      labels:
```

```

        app: web-evs
    spec:
      containers:
      - name: container-1
        image: nginx:latest
        volumeMounts:
        - name: pvc-disk      # Volume name, which must be the same as the
          mountPath: /data  # Location where the storage volume is mounted.
      imagePullSecrets:
      - name: default-secret
      volumes:
      - name: pvc-disk      # Volume name, which can be customized.
        persistentVolumeClaim:
          claimName: pvc-evs  # Name of the created PVC.
    ---
    apiVersion: v1
    kind: Service
    metadata:
      name: web-evs      # Headless Service name.
      namespace: default
      labels:
        app: web-evs
    spec:
      selector:
        app: web-evs
      clusterIP: None
      ports:
      - name: web-evs
        targetPort: 80
        nodePort: 0
        port: 80
        protocol: TCP
      type: ClusterIP
    
```

2. Ejecute el siguiente comando para crear una carga de trabajo en la que está montado el volumen de EVS:

```
kubectl apply -f web-evs.yaml
```

Una vez creada la carga de trabajo, puede probar [Verificación de la persistencia de datos](#).

----Fin

## Verificación de la persistencia de datos

**Paso 1** Vea la aplicación desplegada y los archivos de volumen de EVS.

1. Ejecute el siguiente comando para ver el pod creado:

```
kubectl get pod | grep web-evs
```

Producto esperado:

```
web-evs-0          1/1      Running   0          38s
```

2. Ejecute el siguiente comando para comprobar si el volumen de EVS se ha montado en la ruta **/data**:

```
kubectl exec web-evs-0 -- df | grep data
```

Producto esperado:

```
/dev/sdc          10255636  36888  10202364  0% /data
```

3. Ejecute el siguiente comando para ver los archivos en la ruta **/data**:

```
kubectl exec web-evs-0 -- ls /data
```

Producto esperado:

```
lost+found
```

**Paso 2** Ejecute el siguiente comando para crear un archivo llamado **static** en la ruta **/data**:

```
kubectl exec web-evs-0 -- touch /data/static
```

**Paso 3** Ejecute el siguiente comando para ver los archivos en la ruta **/data**:

```
kubectl exec web-evs-0 -- ls /data
```

Producto esperado:

```
lost+found  
static
```

**Paso 4** Ejecute el siguiente comando para eliminar el pod llamado **web-evs-0**:

```
kubectl delete pod web-evs-0
```

Producto esperado:

```
pod "web-evs-0" deleted
```

**Paso 5** Después de la eliminación, el controlador de StatefulSet crea automáticamente una réplica con el mismo nombre. Ejecute el siguiente comando para comprobar si se han modificado los archivos de la ruta **/data**:

```
kubectl exec web-evs-0 -- ls /data
```

Producto esperado:

```
lost+found  
static
```

Si el archivo **static** todavía existe, los datos en el volumen de EVS se pueden almacenar de forma persistente.

----Fin

## Operaciones relacionadas

También puede realizar las operaciones que aparecen en [Tabla 8-9](#).

**Tabla 8-9** Operaciones relacionadas

Operación	Descripción	Procedimiento
Creación de un volumen de almacenamiento (PV)	Cree un PV en la consola de CCE.	<ol style="list-style-type: none"> <li>1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumes (PVs)</b>. Haga clic en <b>Create Volume</b> en la esquina superior derecha. En el cuadro de diálogo que se muestra, configure los parámetros.                     <ul style="list-style-type: none"> <li>● <b>Volume Type:</b> Seleccione <b>EVS</b>.</li> <li>● <b>EVS:</b> Haga clic en <b>Select EVS</b>. En la página mostrada, seleccione el disco de EVS que cumpla con sus requisitos y haga clic en <b>OK</b>.</li> <li>● <b>PV Name:</b> Introduzca el nombre de PV, que debe ser único en el mismo clúster.</li> <li>● <b>Access Mode:</b> los discos de EVS solo admiten <b>ReadWriteOnce</b>, lo que indica que un volumen de almacenamiento se puede montar en un nodo en modo lectura/escritura. Para obtener más información, véase <a href="#">Modos de acceso al volumen</a>.</li> <li>● <b>Reclaim Policy:</b> <b>Delete</b> o <b>Retain</b>. Para obtener más información, véase <a href="#">Política de reclamo de PV</a>.</li> </ul> </li> <li>2. Haga clic en <b>Create</b>.</li> </ol>
Ampliación de la capacidad de un disco de EVS	<p>Amplíe rápidamente la capacidad de un disco de EVS montado en la consola de CCE.</p> <p>Solo se puede ampliar la capacidad de los discos de EVS de pago por uso en la consola de CCE. Para ampliar la capacidad de los discos de EVS anuales/mensuales, haga clic en el nombre del volumen para ir a la consola de EVS.</p>	<ol style="list-style-type: none"> <li>1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumeClaims (PVCs)</b>. Haga clic en <b>More</b> en la columna <b>Operation</b> del PVC de destino y seleccione <b>Scale-out</b>.</li> <li>2. Ingrese la capacidad que desea agregar y haga clic en <b>OK</b>.</li> </ol>

Operación	Descripción	Procedimiento
Consulta de eventos	Puede ver los nombres de eventos, los tipos de eventos, el número de ocurrencias, los eventos de Kubernetes, la hora de primera ocurrencia y la hora de última ocurrencia del PVC o PV.	<ol style="list-style-type: none"> <li>1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumeClaims (PVCs)</b> o <b>PersistentVolumes (PVs)</b>.</li> <li>2. Haga clic en <b>View Events</b> en la columna <b>Operation</b> del PVC o del PV de destino para ver los eventos generados en una hora (los datos de eventos se conservan durante una hora).</li> </ol>
Consulta de un archivo YAML	Puede ver, copiar y descargar los archivos YAML de un PVC o PV.	<ol style="list-style-type: none"> <li>1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumeClaims (PVCs)</b> o <b>PersistentVolumes (PVs)</b>.</li> <li>2. Haga clic en <b>View YAML</b> en la columna <b>Operation</b> del PVC o del PV de destino para ver o descargar el YAML.</li> </ol>

### 8.3.3 Uso de un disco de EVS con un PV dinámico

Esta sección describe cómo crear automáticamente un disco de EVS y el PV correspondiente. Es aplicable cuando no hay ningún volumen de almacenamiento subyacente disponible.

#### Requisitos previos

- Ha creado un clúster e instalado el complemento de CSI (**everest**) en el clúster.
- Si desea crear un clúster mediante comandos, utilice `kubectrl` para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectrl](#).

#### (Consola) Creación automática de un disco de EVS

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** Cree dinámicamente un PVC y un PV.

1. Elija **Storage** en el panel de navegación y haga clic en la ficha **PersistentVolumeClaims (PVCs)**. Haga clic en **Create PVC** en la esquina superior derecha. En el cuadro de diálogo que se muestra, configure los parámetros de PVC.

Parámetro	Descripción
PVC Type	En este ejemplo, seleccione <b>EVS</b> .
PVC Name	Escriba el nombre de PVC, que debe ser único en el mismo espacio de nombres.

Parámetro	Descripción
Creation Method	<ul style="list-style-type: none"> <li>– Si no hay almacenamiento subyacente disponible, puede seleccionar <b>Dynamically provision</b> para crear un almacenamiento de PVC, PV y subyacente en la consola en modo en cascada.</li> <li>– Si el almacenamiento subyacente está disponible, puede crear un volumen de almacenamiento o utilizar un volumen de almacenamiento existente para crear estáticamente un PVC en función de si se ha creado un PV. Para obtener más información, véase <a href="#">Uso de un disco de EVS existente a través de un PV estático</a>.</li> </ul> <p>En este ejemplo, seleccione <b>Dynamically provision</b>.</p>
Storage Classes	La clase de almacenamiento para los discos de EVS es <b>csi-disk</b> .
AZ	<p>Seleccione la AZ del disco de EVS. La AZ debe ser la misma que la del nodo del clúster.</p> <p><b>NOTA</b>                      Un disco de EVS solo se puede montar en un nodo en la misma AZ. Después de crear un disco de EVS, no se puede cambiar su AZ.</p>
Disk Type	Seleccione un tipo de disco de EVS.
Access Mode	Los discos de EVS solo admiten <b>ReadWriteOnce</b> , lo que indica que un volumen de almacenamiento se puede montar en un nodo en modo de lectura/escritura. Para obtener más información, véase <a href="#">Modos de acceso al volumen</a> .
Capacity (GiB)	Capacidad del volumen de almacenamiento solicitado.
Encryption	Puede seleccionar <b>Encryption</b> y una clave de encriptación para cifrar el almacenamiento subyacente. Solo los discos de EVS y los sistemas de archivos SFS admiten encriptación.
Enterprise Project	Proyectos de empresa admitidos: predeterminado, al que pertenece el clúster o al que especifica la clase de almacenamiento.

2. Haga clic en **Create**.

Puede elegir **Storage** en el panel de navegación y ver el PVC y PV creados en las páginas de fichas **PersistentVolumeClaims (PVCs)** y **PersistentVolumes (PVs)**.

**Paso 3** Cree una aplicación.

1. En el panel de navegación de la izquierda, haga clic en **Workloads**. En el panel derecho, haga clic en la ficha **StatefulSets**.
2. Haga clic en **Create Workload** en la esquina superior derecha. En la página mostrada, haga clic en **Data Storage** en el área **Container Settings** y haga clic en **Add Volume** para seleccionar **PVC**.

Montar y utilizar volúmenes de almacenamiento, como se muestra en [Tabla 8-10](#). Para obtener más información sobre otros parámetros, consulte [Workloads](#).

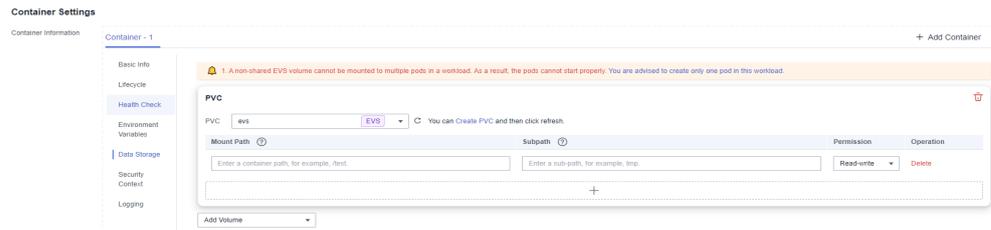
**Tabla 8-10** Montaje de un volumen de almacenamiento

Parámetro	Descripción
PVC	<p>Seleccione un volumen de EVS existente.</p> <p>Un volumen de EVS no se puede montar repetidamente en varias cargas de trabajo.</p>
Add Container Path	<p>Configure los parámetros siguientes:</p> <ol style="list-style-type: none"> <li> <p><b>Mount Path:</b> Introduzca una ruta de acceso del contenedor. Por ejemplo, <code>/tmp</code>. Este parámetro indica la ruta de contenedor en la que se montará un volumen de datos. No monte el volumen en un directorio del sistema como <code>/</code> o <code>/var/run</code>; esta acción puede causar errores de contenedor. Se recomienda montar el volumen en un directorio vacío. Si el directorio no está vacío, asegúrese de que no hay archivos que afecten al inicio de contenedor. De lo contrario, se reemplazarán los archivos, lo que provocará errores de inicio de contenedor o errores de creación de carga de trabajo.</p> <p><b>AVISO</b></p> <p>Cuando el contenedor está montado en un directorio de alto riesgo, se recomienda utilizar una cuenta con los permisos mínimos para iniciar el contenedor; de lo contrario, los archivos de alto riesgo en la máquina host podrían estar dañados.</p> </li> <li> <p><b>Subpath:</b> Introduzca una ruta secundaria, por ejemplo, <code>tmp</code>. Se utiliza una ruta secundaria para montar un volumen local de modo que se utilice el mismo volumen de datos en un solo pod. Si este parámetro se deja en blanco, la ruta raíz se utiliza de forma predeterminada.</p> </li> <li> <p><b>Permission</b></p> <ul style="list-style-type: none"> <li> <p><b>Read-only:</b> Solo puede leer los datos en los volúmenes montados.</p> </li> <li> <p><b>Read/Write:</b> Puede modificar los volúmenes de datos montados en la ruta de acceso. Los datos recién escritos no se migran si se migra el contenedor, lo que puede causar la pérdida de datos.</p> </li> </ul> </li> </ol> <p>Puede hacer clic en  para agregar varias rutas y subrutas.</p>

En este ejemplo, el disco se monta en la trayectoria `/data` del contenedor. Los datos de contenedor generados en esta ruta se almacenan en el disco de EVS.

 **NOTA**

Un disco ReadWriteOnce de EVS no compartido no se puede montar en varios pods de una carga de trabajo. Como resultado, los pods no pueden arrancar correctamente. Asegúrese de que el número de pods de carga de trabajo es 1 al montar un disco de EVS.



- Después de completar la configuración, haga clic en **Create**.  
Una vez creada la carga de trabajo, puede probar **Verificación de la persistencia de datos**.

----Fin

## (kubect) Creación automática de un disco de EVS

**Paso 1** Utilice kubectl para conectarse al clúster.

**Paso 2** Utilice StorageClass para crear dinámicamente un PVC y un PV.

- Cree el archivo **pvc-evs-auto.yaml**.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-evs-auto
  namespace: default
  annotations:
    everest.io/disk-volume-type: SAS # EVS disk type.
    everest.io/crypt-key-id: <your_key_id> # (Optional) Encryption key ID.
    everest.io/enterprise-project-id: <your_project_id> # (Optional)
Mandatory for an encrypted disk.
Enterprise project ID. If an enterprise project is specified, you need to use
the same enterprise project when creating a PVC. Otherwise, the PVC cannot be
bound to a PV.
  labels:
    failure-domain.beta.kubernetes.io/region: <your_region> # Region of the
node where the application is to be deployed.
    failure-domain.beta.kubernetes.io/zone: <your_zone> # AZ of the
node where the application is to be deployed.
spec:
  accessModes:
    - ReadWriteOnce # The value must be ReadWriteOnce for EVS
disks.
  resources:
    requests:
      storage: 10Gi # EVS disk capacity, ranging from 1 to 32768.
      storageClassName: csi-disk # Storage class type for EVS disks.
```

**Tabla 8-11** Parámetros clave

Parámetro	Obligatorio	Descripción
failure-domain.beta.kubernetes.io/region	Sí	Región donde se encuentra el clúster. Para obtener más información sobre el valor de <b>region</b> , consulte <b>Regiones y puntos de conexión</b> .

Parámetro	Obligatorio	Descripción
failure-domain.beta.kubernetes.io/zone	Sí	AZ donde se crea el volumen de EVS. Debe ser la misma que la AZ prevista para la carga de trabajo. Para obtener más información sobre el valor de <b>zone</b> , consulte <a href="#">Regiones y puntos de conexión</a> .
everest.io/disk-volume-type	Sí	Tipo del disco de EVS. Todas las letras están en mayúsculas. <ul style="list-style-type: none"> <li>– <b>SAS</b>: E/S con capacidad alta</li> <li>– <b>SSD</b>: E/S con capacidad ultraalta</li> <li>– <b>GPSSD</b>: SSD de uso general</li> <li>– <b>ESSD</b>: SSD extremo</li> </ul>
everest.io/crypt-key-id	No	Obligatorio cuando el disco de EVS está cifrado. Introduzca el ID de clave de encriptación seleccionado durante la creación del disco de EVS. Para obtener el ID de clave de encriptación, inicie sesión en <b>Cloud Server Console</b> . En el panel de navegación, elija <b>Elastic Volume Service &gt; Disks</b> . Haga clic en el nombre del disco de EVS de destino para ir a su página de detalles. En la página de ficha <b>Summary</b> , copie el valor de <b>KMS Key ID</b> en el área <b>Configuration Information</b> .
everest.io/enterprise-project-id	No	Opcional. ID de proyecto de empresa del disco de EVS. Si se especifica un proyecto de empresa, debe especificar el mismo proyecto de empresa al crear un PVC. De lo contrario, el PVC no puede estar unido a un PV. Para obtener el ID del proyecto de empresa, inicie sesión en <b>Cloud Server Console</b> . En el panel de navegación, elija <b>Elastic Volume Service &gt; Disks</b> . Haga clic en el nombre del disco de EVS de destino para ir a su página de detalles. En la página de ficha <b>Summary</b> , haga clic en el proyecto de empresa de <b>Management Information</b> para acceder a la consola del proyecto de empresa. Copie el ID correspondiente para obtener el ID del proyecto de empresa al que pertenece el disco de EVS.
storage	Sí	Capacidad de PVC solicitada, en Gi. El valor oscila entre <b>1</b> y <b>32768</b> .
storageClassName	Sí	El nombre de clase de almacenamiento de los volúmenes de EVS es <b>csi-disk</b> .

2. Ejecute el siguiente comando para crear un PVC:

```
kubectl apply -f pvc-evs-auto.yaml
```

### Paso 3 Cree una aplicación.

1. Cree un archivo denominado **web-evs-auto.yaml**. En este ejemplo, el volumen de EVS se monta en la ruta **/data**.

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: web-evs-auto
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      app: web-evs-auto
  serviceName: web-evs-auto # Headless Service name.
  template:
    metadata:
      labels:
        app: web-evs-auto
    spec:
      containers:
        - name: container-1
          image: nginx:latest
          volumeMounts:
            - name: pvc-disk # Volume name, which must be the same as the
              volume name in the volumes field.
              mountPath: /data # Location where the storage volume is mounted.
          imagePullSecrets:
            - name: default-secret
      volumes:
        - name: pvc-disk # Volume name, which can be customized.
          persistentVolumeClaim:
            claimName: pvc-evs-auto # Name of the created PVC.
---
apiVersion: v1
kind: Service
metadata:
  name: web-evs-auto # Headless Service name.
  namespace: default
  labels:
    app: web-evs-auto
spec:
  selector:
    app: web-evs-auto
  clusterIP: None
  ports:
    - name: web-evs-auto
      targetPort: 80
      nodePort: 0
      port: 80
      protocol: TCP
  type: ClusterIP
```

2. Ejecute el siguiente comando para crear una carga de trabajo en la que está montado el volumen de EVS:

```
kubectl apply -f web-evs-auto.yaml
```

Una vez creada la carga de trabajo, puede probar [Verificación de la persistencia de datos](#).

----Fin

## Verificación de la persistencia de datos

**Paso 1** Vea la aplicación desplegada y los archivos de volumen de EVS.

1. Ejecute el siguiente comando para ver el pod creado:

```
kubectl get pod | grep web-evs-auto
```

Producto esperado:

```
web-evs-auto-0          1/1      Running   0          38s
```

2. Run the following command to check whether the EVS volume has been mounted to the **/data** path:

```
kubectl exec web-evs-auto-0 -- df | grep data
```

Producto esperado:

```
/dev/sdc                10255636   36888   10202364   0% /data
```

3. Ejecute el siguiente comando para ver los archivos en la ruta **/data**:

```
kubectl exec web-evs-auto-0 -- ls /data
```

Producto esperado:

```
lost+found
```

**Paso 2** Ejecute el siguiente comando para crear un archivo llamado **static** en la ruta **/data**:

```
kubectl exec web-evs-auto-0 -- touch /data/static
```

**Paso 3** Ejecute el siguiente comando para ver los archivos en la ruta **/data**:

```
kubectl exec web-evs-auto-0 -- ls /data
```

Producto esperado:

```
lost+found
static
```

**Paso 4** Ejecute el siguiente comando para eliminar el pod llamado **web-evs-auto-0**:

```
kubectl delete pod web-evs-auto-0
```

Producto esperado:

```
pod "web-evs-auto-0" deleted
```

**Paso 5** Después de la eliminación, el controlador de StatefulSet crea automáticamente una réplica con el mismo nombre. Ejecute el siguiente comando para comprobar si se han modificado los archivos de la ruta **/data**:

```
kubectl exec web-evs-auto-0 -- ls /data
```

Producto esperado:

```
lost+found
static
```

Si el archivo **static** todavía existe, los datos en el volumen de EVS se pueden almacenar de forma persistente.

----Fin

## Operaciones relacionadas

También puede realizar las operaciones que aparecen en [Tabla 8-12](#).

**Tabla 8-12** Operaciones relacionadas

Operación	Descripción	Procedimiento
Ampliación de la capacidad de un disco de EVS	Amplíe rápidamente la capacidad de un disco de EVS montado en la consola de CCE.  Solo se puede ampliar la capacidad de los discos de EVS de pago por uso en la consola de CCE. Para ampliar la capacidad de los discos de EVS anuales/mensuales, haga clic en el nombre del volumen para ir a la consola de EVS.	<ol style="list-style-type: none"> <li>1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumeClaims (PVCs)</b>. Haga clic en <b>More</b> en la columna <b>Operation</b> del PVC de destino y seleccione <b>Scale-out</b>.</li> <li>2. Ingrese la capacidad que desea agregar y haga clic en <b>OK</b>.</li> </ol>
Consulta de eventos	Puede ver los nombres de eventos, los tipos de eventos, el número de ocurrencias, los eventos de Kubernetes, la hora de primera ocurrencia y la hora de última ocurrencia del PVC o PV.	<ol style="list-style-type: none"> <li>1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumeClaims (PVCs)</b> o <b>PersistentVolumes (PVs)</b>.</li> <li>2. Haga clic en <b>View Events</b> en la columna <b>Operation</b> del PVC o del PV de destino para ver los eventos generados en una hora (los datos de eventos se conservan durante una hora).</li> </ol>
Consulta de un archivo YAML	Puede ver, copiar y descargar los archivos YAML de un PVC o PV.	<ol style="list-style-type: none"> <li>1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumeClaims (PVCs)</b> o <b>PersistentVolumes (PVs)</b>.</li> <li>2. Haga clic en <b>View YAML</b> en la columna <b>Operation</b> del PVC o del PV de destino para ver o descargar el YAML.</li> </ol>

### 8.3.4 Montaje dinámico de un disco de EVS en un StatefulSet

#### Escenarios de aplicación

El montaje dinámico solo está disponible para crear un **StatefulSet**. Se implementa con una plantilla de reclamo de volumen (campo **volumeClaimTemplates**) y depende de la clase de almacenamiento para aprovisionar PV dinámicamente. En este modo, cada pod en un StatefulSet de múltiples pods está asociado con un PVC y un PV únicos. Después de reprogramar un pod, los datos originales todavía se pueden montar en él basándose en el nombre de PVC. En el modo de montaje común para una Deployment, si se admite ReadWriteMany, varios pods de la Deployment se montarán en el mismo almacenamiento subyacente.

## Requisitos previos

- Ha creado un clúster e instalado el complemento de CSI (**everest**) en el clúster.
- Si desea crear un clúster mediante comandos, utilice kubectl para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).

## (Consola) Montaje dinámico de un disco de EVS

- Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.
- Paso 2** En el panel de navegación de la izquierda, haga clic en **Workloads**. En el panel derecho, haga clic en la ficha **StatefulSets**.
- Paso 3** Haga clic en **Create Workload** en la esquina superior derecha. En la página mostrada, haga clic en **Data Storage** en el área **Container Settings** y haga clic en **Add Volume** para seleccionar **VolumeClaimTemplate (VTC)**.
- Paso 4** Haga clic en **Create PVC**. En el cuadro de diálogo que se muestra, configure los parámetros de plantilla de notificación de volumen.

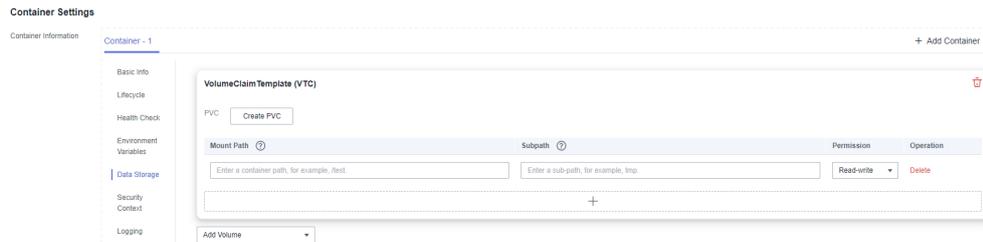
Montar y utilizar dinámicamente volúmenes de almacenamiento. Para obtener más información sobre otros parámetros, consulte [Creación de un StatefulSet](#).

Parámetro	Descripción
PVC Type	En este ejemplo, seleccione <b>EVS</b> .
PVC Name	Escriba el nombre del PVC. Después de crear un PVC, se agrega automáticamente un sufijo en función del número de instancias. El formato es <i>&lt;Custom PVC name&gt;-&lt;Serial number&gt;</i> , por ejemplo, <i>example-0</i> .
Creation Method	Puede seleccionar <b>Dynamically provision</b> para crear un PVC, PV y almacenamiento subyacente en la consola en modo en cascada.
Storage Classes	La clase de almacenamiento para los discos de EVS es <b>csi-disk</b> .
AZ	Seleccione la AZ del disco de EVS. La AZ debe ser la misma que la del nodo del clúster. <b>NOTA</b> Un disco de EVS solo se puede montar en un nodo en la misma AZ. Después de crear un disco de EVS, no se puede cambiar su AZ.
Disk Type	Seleccione un tipo de disco de EVS.
Access Mode	Los discos de EVS solo admiten <b>ReadWriteOnce</b> , lo que indica que un volumen de almacenamiento se puede montar en un nodo en modo de lectura/escritura. Para obtener más información, véase <a href="#">Modos de acceso al volumen</a> .
Capacity (GiB)	Capacidad del volumen de almacenamiento solicitado.
Encryption	Puede seleccionar <b>Encryption</b> y una clave de encriptación para cifrar el almacenamiento subyacente. Solo los discos de EVS y los sistemas de archivos SFS admiten encriptación.

Parámetro	Descripción
Enterprise Project	Proyectos de empresa admitidos: predeterminado, al que pertenece el clúster o al que especifica la clase de almacenamiento.

**Paso 5** Introduzca la ruta en la que está montado el volumen.

En este ejemplo, el disco se monta en la trayectoria **/data** del contenedor. Los datos de contenedor generados en esta ruta se almacenan en el disco de EVS.



**Paso 6** Después de completar la configuración, haga clic en **Create**.

Una vez creada la carga de trabajo, puede probar [Verificación de la persistencia de datos](#).

----Fin

## (kubectl) Uso de un disco de EVS existente

**Paso 1** Utilice kubectl para conectarse al clúster.

**Paso 2** Cree un archivo denominado **statefulset-evs.yaml**. En este ejemplo, el volumen de EVS se monta en la ruta **/data**.

```

apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: statefulset-evs
  namespace: default
spec:
  selector:
    matchLabels:
      app: statefulset-evs
  template:
    metadata:
      labels:
        app: statefulset-evs
    spec:
      containers:
        - name: container-1
          image: nginx:latest
          volumeMounts:
            - name: pvc-disk # The value must be the same as that in
              # Location where the storage volume is
              mountPath: /data
      volumeClaimTemplates:
        - apiVersion: v1
          kind: PersistentVolumeClaim
          metadata:
  
```

```

name: pvc-disk
namespace: default
annotations:
  everest.io/disk-volume-type: SAS # EVS disk type.
  everest.io/crypt-key-id: <your_key_id> # (Optional) Encryption key
  everest.io/enterprise-project-id: <your_project_id> # (Optional)
  ID. Mandatory for an encrypted disk.
  Enterprise project ID. If an enterprise project is specified, you need to use the
  same enterprise project when creating a PVC. Otherwise, the PVC cannot be bound
  to a PV.
labels:
  failure-domain.beta.kubernetes.io/region: <your_region> # Region of
  the node where the application is to be deployed.
  failure-domain.beta.kubernetes.io/zone: <your_zone> # AZ of the
  node where the application is to be deployed.
spec:
  accessModes:
    - ReadWriteOnce # The value must be ReadWriteOnce for EVS
  disks.
  resources:
    requests:
      storage: 10Gi # EVS disk capacity, ranging from 1 to
      32768.
      storageClassName: csi-disk # Storage class type for EVS disks.
  ---
apiVersion: v1
kind: Service
metadata:
  name: statefulset-evs # Headless Service name.
  namespace: default
  labels:
    app: statefulset-evs
spec:
  selector:
    app: statefulset-evs
  clusterIP: None
  ports:
    - name: statefulset-evs
      targetPort: 80
      nodePort: 0
      port: 80
      protocol: TCP
  type: ClusterIP
    
```

**Tabla 8-13** Parámetros clave

Parámetro	Obligatorio	Descripción
failure-domain.beta.kubernetes.io/region	Sí	Región donde se encuentra el clúster. Para obtener más información sobre el valor de <b>region</b> , consulte <a href="#">Regiones y puntos de conexión</a> .
failure-domain.beta.kubernetes.io/zone	Sí	AZ donde se crea el volumen de EVS. Debe ser la misma que la AZ prevista para la carga de trabajo. Para obtener más información sobre el valor de <b>zone</b> , consulte <a href="#">Regiones y puntos de conexión</a> .

Parámetro	Obligatorio	Descripción
everest.io/disk-volume-type	Sí	Tipo del disco de EVS. Todas las letras están en mayúsculas. <ul style="list-style-type: none"> <li>● <b>SAS</b>: E/S con capacidad alta</li> <li>● <b>SSD</b>: E/S con capacidad ultraalta</li> <li>● <b>GPSSD</b>: SSD de uso general</li> <li>● <b>ESSD</b>: SSD extremo</li> </ul>
everest.io/crypt-key-id	No	Obligatorio cuando el disco de EVS está cifrado. Introduzca el ID de clave de encriptación seleccionado durante la creación del disco de EVS. Para obtener el ID de clave de encriptación, inicie sesión en <b>Cloud Server Console</b> . En el panel de navegación, elija <b>Elastic Volume Service &gt; Disks</b> . Haga clic en el nombre del disco de EVS de destino para ir a su página de detalles. En la página de ficha <b>Summary</b> , copie el valor de <b>KMS Key ID</b> en el área <b>Configuration Information</b> .
everest.io/enterprise-project-id	No	Opcional. ID de proyecto de empresa del disco de EVS. Si se especifica un proyecto de empresa, debe especificar el mismo proyecto de empresa al crear un PVC. De lo contrario, el PVC no puede estar unido a un PV. Para obtener el ID del proyecto de empresa, inicie sesión en <b>Cloud Server Console</b> . En el panel de navegación, elija <b>Elastic Volume Service &gt; Disks</b> . Haga clic en el nombre del disco de EVS de destino para ir a su página de detalles. En la página de ficha <b>Summary</b> , haga clic en el proyecto de empresa de <b>Management Information</b> para acceder a la consola del proyecto de empresa. Copie el ID correspondiente para obtener el ID del proyecto de empresa al que pertenece el disco de EVS.
storage	Sí	Capacidad de PVC solicitada, en Gi. El valor oscila entre <b>1</b> y <b>32768</b> .
storageClassName	Sí	El nombre de clase de almacenamiento para los discos de EVS es <b>csi-disk</b> .

**Paso 3** Ejecute el siguiente comando para crear una carga de trabajo en la que está montado el volumen de EVS:

```
kubectl apply -f statefulset-evs.yaml
```

Una vez creada la carga de trabajo, puede probar [Verificación de la persistencia de datos](#).

----Fin

## Verificación de la persistencia de datos

**Paso 1** Vea la aplicación desplegada y los archivos de volumen de EVS.

1. Ejecute el siguiente comando para ver el pod creado:

```
kubectl get pod | grep statefulset-evs
```

Producto esperado:

```
statefulset-evs-0          1/1      Running    0          45s
statefulset-evs-1          1/1      Running    0          28s
```

2. Ejecute el siguiente comando para comprobar si el volumen de EVS se ha montado en la ruta **/data**:

```
kubectl exec statefulset-evs-0 -- df | grep data
```

Producto esperado:

```
/dev/sdd          10255636      36888 10202364   0% /data
```

3. Ejecute el siguiente comando para ver los archivos en la ruta **/data**:

```
kubectl exec statefulset-evs-0 -- ls /data
```

Producto esperado:

```
lost+found
```

**Paso 2** Ejecute el siguiente comando para crear un archivo llamado **static** en la ruta **/data**:

```
kubectl exec statefulset-evs-0 -- touch /data/static
```

**Paso 3** Ejecute el siguiente comando para ver los archivos en la ruta **/data**:

```
kubectl exec statefulset-evs-0 -- ls /data
```

Producto esperado:

```
lost+found
static
```

**Paso 4** Ejecute el siguiente comando para eliminar el pod llamado **web-evs-auto-0**:

```
kubectl delete pod statefulset-evs-0
```

Producto esperado:

```
pod "statefulset-evs-0" deleted
```

**Paso 5** Después de la eliminación, el controlador de StatefulSet crea automáticamente una réplica con el mismo nombre. Ejecute el siguiente comando para comprobar si se han modificado los archivos de la ruta **/data**:

```
kubectl exec statefulset-evs-0 -- ls /data
```

Producto esperado:

```
lost+found
static
```

Si el archivo **static** todavía existe, los datos en el volumen de EVS se pueden almacenar de forma persistente.

----Fin

## Operaciones relacionadas

También puede realizar las operaciones que aparecen en [Tabla 8-14](#).

**Tabla 8-14** Operaciones relacionadas

Operación	Descripción	Procedimiento
Ampliación de la capacidad de un disco de EVS	Amplíe rápidamente la capacidad de un disco de EVS montado en la consola de CCE.  Solo se puede ampliar la capacidad de los discos de EVS de pago por uso en la consola de CCE. Para ampliar la capacidad de los discos de EVS anuales/mensuales, haga clic en el nombre del volumen para ir a la consola de EVS.	<ol style="list-style-type: none"> <li>1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumeClaims (PVCs)</b>. Haga clic en <b>More</b> en la columna <b>Operation</b> del PVC de destino y seleccione <b>Scale-out</b>.</li> <li>2. Ingrese la capacidad que desea agregar y haga clic en <b>OK</b>.</li> </ol>
Consulta de eventos	Puede ver los nombres de eventos, los tipos de eventos, el número de ocurrencias, los eventos de Kubernetes, la hora de primera ocurrencia y la hora de última ocurrencia del PVC o PV.	<ol style="list-style-type: none"> <li>1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumeClaims (PVCs)</b> o <b>PersistentVolumes (PVs)</b>.</li> <li>2. Haga clic en <b>View Events</b> en la columna <b>Operation</b> del PVC o del PV de destino para ver los eventos generados en una hora (los datos de eventos se conservan durante una hora).</li> </ol>
Consulta de un archivo YAML	Puede ver, copiar y descargar los archivos YAML de un PVC o PV.	<ol style="list-style-type: none"> <li>1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumeClaims (PVCs)</b> o <b>PersistentVolumes (PVs)</b>.</li> <li>2. Haga clic en <b>View YAML</b> en la columna <b>Operation</b> del PVC o del PV de destino para ver o descargar el YAML.</li> </ol>

## 8.4 Scalable File Service (SFS)

### 8.4.1 Descripción general

#### Presentación

CCE permite montar un volumen creado a partir de un sistema de archivos Scalable File Service (SFS) en un contenedor para almacenar datos de forma persistente. Los volúmenes de SFS se usan comúnmente en escenarios de ReadWriteMany para servicios de expansión de gran capacidad y costos sensibles, como procesamiento de medios, gestión de contenido, análisis de big data y análisis de procesos de carga de trabajo. Para los servicios con un

volumen masivo de archivos pequeños, se recomiendan los sistemas de archivos de SFS Turbo.

Ampliable a petabytes, SFS proporciona almacenamiento de archivos compartidos totalmente alojado, altamente disponible y estable para manejar aplicaciones de uso intensivo de datos y ancho de banda

- **Standard file protocols:** Puede montar sistemas de archivos como volúmenes en servidores, lo mismo que usar directorios locales.
- **Data sharing:** El mismo sistema de archivos se puede montar en varios servidores, para que los datos se puedan compartir.
- **Private network:** El usuario solo puede acceder a los datos en redes privadas de centros de datos.
- **Capacity and performance:** La capacidad de un solo sistema de archivos es alta (nivel PB) y el rendimiento es excelente (latencia de E/S de lectura/escritura a nivel ms).
- **Use cases:** Deployments/StatefulSets en modo ReadWriteMany y trabajos creados para cómputo de alto rendimiento (HPC), procesamiento de medios, gestión de contenido, servicios web, análisis de big data y análisis de procesos de carga de trabajo

## Rendimiento

CCE admite sistemas de archivos de SFS Capacity-Oriented y de SFS 3.0 Capacity-Oriented. Para obtener más información sobre los tipos de sistema de archivos, consulte [Tipo de sistema de archivo](#).

### NOTA

- Los sistemas de archivos de SFS Capacity-Oriented están agotados y no se pueden crear en la consola de CCE. Todavía puede crear PV para los sistemas de archivos existentes de SFS Capacity-Oriented con [kubectl](#).

**Tabla 8-15** Rendimiento

Parámetro	SFS Capacity-Oriented
Ancho de banda máximo	2 GB/s
IOPS máximas	2,000
Latencia	3–20 ms
Capacidad máxima	4 PB

## Escenarios de aplicación

SFS admite los siguientes modos de montaje basados en escenarios de aplicación:

- **Uso de un sistema de archivos de SFS existente con un PV estático:** modo de creación estática, en el que se utiliza un volumen de SFS existente para crear un PV y, a continuación, montar el almacenamiento en la carga de trabajo con un PVC. Este modo es aplicable a escenarios en los que el almacenamiento subyacente está disponible o se factura anualmente/mensualmente.

- **Uso de un sistema de archivos SFS a través de un PV dinámico:** modo de creación dinámica, donde no es necesario crear volúmenes de SFS por adelantado. En su lugar, especifique un StorageClass durante la creación de PVC y se creará automáticamente un volumen SFS y un PV. Este modo es aplicable a escenarios en los que no hay almacenamiento subyacente disponible.

## Restricciones

- El sistema de archivos de SFS y el clúster deben estar en la misma VPC.
- Varios PV pueden utilizar el mismo sistema de archivos SFS o SFS Turbo con las siguientes restricciones:
  - Puede producirse un error si varios PVC/PV que utilizan el mismo sistema de archivos SFS o SFS Turbo subyacente están montados en el mismo pod.
  - El parámetro **persistentVolumeReclaimPolicy** de los PV debe establecerse en **Retain**. De lo contrario, cuando se elimina un PV, se puede eliminar el volumen subyacente asociado. En este caso, otros PV asociados con el volumen subyacente pueden ser anormales.
  - Cuando el volumen subyacente se utiliza repetidamente, se recomienda desplegar ReadWriteMany en la capa de aplicación para evitar la sobrescritura y la pérdida de datos.

## Facturación

- Para los volúmenes de SFS **creados automáticamente** con StorageClass, el modo de facturación predeterminado es **Pay-per-use**, lo que indica que se cobra en función de la capacidad de almacenamiento y la duración utilizadas. Para obtener más información sobre los precios de SFS, consulte **Facturación**.
- Si desea que se le facture en modo anual/mensual, **utilice los volúmenes de archivos de SFS existentes**.

## 8.4.2 Uso de un sistema de archivos de SFS existente con un PV estático

SFS es un almacenamiento conectado a la red (NAS) que proporciona almacenamiento de archivos compartido, escalable y de alto rendimiento. Es aplicable a la expansión de gran capacidad y a los servicios sensibles a los costos. Esta sección describe cómo utilizar un sistema de archivos SFS existente para crear estáticamente los PV y PVC e implementar la persistencia de datos y el uso compartido en las cargas de trabajo.

### Requisitos previos

- Ha creado un clúster e instalado el complemento de CSI (**everest**) en el clúster.
- Si desea crear un clúster mediante comandos, utilice kubectl para conectarse al clúster. Para obtener más información, véase **Conexión a un clúster con kubectl**.
- Ha creado un sistema de archivos SFS que está en la misma VPC que el clúster.

## Restricciones

- El sistema de archivos de SFS y el clúster deben estar en la misma VPC.
- Varios PV pueden utilizar el mismo sistema de archivos SFS o SFS Turbo con las siguientes restricciones:

- Puede producirse un error si varios PVC/PV que utilizan el mismo sistema de archivos SFS o SFS Turbo subyacente están montados en el mismo pod.
- El parámetro **persistentVolumeReclaimPolicy** de los PV debe establecerse en **Retain**. De lo contrario, cuando se elimina un PV, se puede eliminar el volumen subyacente asociado. En este caso, otros PV asociados con el volumen subyacente pueden ser anormales.
- Cuando el volumen subyacente se utiliza repetidamente, se recomienda desplegar ReadWriteMany en la capa de aplicación para evitar la sobreescritura y la pérdida de datos.

## (Consola) Uso de un sistema de archivos de SFS existente

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** Cree un PV y PVC estáticamente.

1. Elija **Storage** en el panel de navegación y haga clic en la ficha **PersistentVolumeClaims (PVCs)**. Haga clic en **Create PVC** en la esquina superior derecha. En el cuadro de diálogo que se muestra, configure los parámetros de PVC.

Parámetro	Descripción
PVC Type	En este ejemplo, seleccione <b>SFS</b> .
PVC Name	Escriba el nombre de PVC, que debe ser único en el mismo espacio de nombres.
Creation Method	<ul style="list-style-type: none"> <li>– Si el almacenamiento subyacente está disponible, puede crear un volumen de almacenamiento o utilizar un volumen de almacenamiento existente para crear estáticamente un PVC en función de si se ha creado un PV.</li> <li>– Si no hay ningún almacenamiento subyacente disponible, puede seleccionar <b>Dynamically provision</b>. Para obtener más información, véase <a href="#">Uso de un sistema de archivos SFS a través de un PV dinámico</a>.</li> </ul> <p>En este ejemplo, seleccione <b>Create new</b> para crear un PV y un PVC al mismo tiempo en la consola.</p>
PV <sup>a</sup>	<p>Seleccione un PV existente en el clúster. Es necesario crear un PV por adelantado. Para obtener más información, consulte "Creación de un volumen de almacenamiento (PV)" de <a href="#">Operaciones relacionadas</a>.</p> <p>En este ejemplo, no es necesario establecer este parámetro.</p>
SFS <sup>b</sup>	<p>Haga clic en <b>Select SFS</b>. En la página mostrada, seleccione el sistema de archivos SFS que cumpla con sus requisitos y haga clic en <b>OK</b>.</p> <p><b>NOTA</b> Actualmente, solo se admite SFS 3.0 Capacity-Oriented.</p>
PV Name <sup>b</sup>	Introduzca el nombre de PV, que debe ser único en el mismo clúster.

Parámetro	Descripción
Access Mode <sup>b</sup>	Los volúmenes de SFS solo admiten <b>ReadWriteMany</b> , lo que indica que un volumen de almacenamiento se puede montar en varios nodos en modo de lectura/escritura. Para obtener más información, véase <a href="#">Modos de acceso al volumen</a> .
Reclaim Policy <sup>b</sup>	Puede seleccionar <b>Delete</b> o <b>Retain</b> para especificar la política de recuperación del almacenamiento subyacente cuando se elimina el PVC. Para obtener más información, véase <a href="#">Política de reclamo de PV</a> .  <b>NOTA</b> Si varios PV utilizan el mismo volumen de almacenamiento subyacente, utilice <b>Retain</b> para evitar la eliminación en cascada de volúmenes subyacentes.
Mount Options <sup>b</sup>	Introduzca los pares de clave-valor del parámetro de montaje. Para obtener más información, véase <a href="#">Configuración de las opciones de montaje de volumen de SFS</a> .

 **NOTA**

- a: El parámetro está disponible cuando **Creation Method** se establece en **Use existing**.
- b: El parámetro está disponible cuando **Creation Method** se establece en **Create new**.

2. Haga clic en **Create** para crear un PVC y un PV.

Puede elegir **Storage** en el panel de navegación y ver el PVC y PV creados en las páginas de fichas **PersistentVolumeClaims (PVCs)** y **PersistentVolumes (PVs)**.

**Paso 3** Cree una aplicación.

1. En el panel de navegación de la izquierda, haga clic en **Workloads**. En el panel derecho, haga clic en la ficha **Deployments**.
2. Haga clic en **Create Workload** en la esquina superior derecha. En la página mostrada, haga clic en **Data Storage** en el área **Container Settings** y haga clic en **Add Volume** para seleccionar **PVC**.

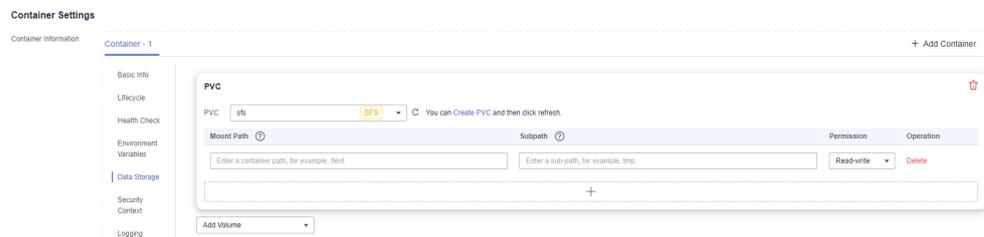
Montar y utilizar volúmenes de almacenamiento, como se muestra en [Tabla 8-16](#). Para obtener más información sobre otros parámetros, consulte [Workloads](#).

**Tabla 8-16** Montaje de un volumen de almacenamiento

Parámetro	Descripción
PVC	Seleccione un volumen de SFS existente.

Parámetro	Descripción
Add Container Path	<p>Configure los parámetros siguientes:</p> <ol style="list-style-type: none"> <li><b>Mount Path:</b> Introduzca una ruta de acceso del contenedor. Por ejemplo, <code>/tmp</code>. Este parámetro indica la ruta de contenedor en la que se montará un volumen de datos. No monte el volumen en un directorio del sistema como <code>/</code> o <code>/var/run</code>; esta acción puede causar errores de contenedor. Se recomienda montar el volumen en un directorio vacío. Si el directorio no está vacío, asegúrese de que no hay archivos que afecten al inicio de contenedor. De lo contrario, se reemplazarán los archivos, lo que provocará errores de inicio de contenedor o errores de creación de carga de trabajo.                     <p><b>AVISO</b></p>                     Cuando el contenedor está montado en un directorio de alto riesgo, se recomienda utilizar una cuenta con los permisos mínimos para iniciar el contenedor; de lo contrario, los archivos de alto riesgo en la máquina host podrían estar dañados.</li> <li><b>Subpath:</b> Introduzca una ruta secundaria, por ejemplo, <code>tmp</code>. Se utiliza una ruta secundaria para montar un volumen local de modo que se utilice el mismo volumen de datos en un solo pod. Si este parámetro se deja en blanco, la ruta raíz se utiliza de forma predeterminada.</li> <li><b>Permission</b> <ul style="list-style-type: none"> <li><b>Read-only:</b> Solo puede leer los datos en los volúmenes montados.</li> <li><b>Read/Write:</b> Puede modificar los volúmenes de datos montados en la ruta de acceso. Los datos recién escritos no se migran si se migra el contenedor, lo que puede causar la pérdida de datos.</li> </ul> </li> </ol> <p>Puede hacer clic en <b>+</b> para agregar varias rutas y subrutas.</p>

En este ejemplo, el disco se monta en la trayectoria `/data` del contenedor. Los datos de contenedor generados en esta ruta se almacenan en el sistema de archivos SFS.



3. Después de completar la configuración, haga clic en **Create**.

Una vez creada la carga de trabajo, puede probar [Verificación de la persistencia y el uso compartido de datos](#).

----Fin

## (kubectl) Uso de un sistema de archivos SFS existente

**Paso 1** Utilice kubectl para conectarse al clúster.

**Paso 2** Crear un PV.

1. Cree el archivo **pv-sfs.yaml**.

SFS Capacity-Oriented:

```
apiVersion: v1
kind: PersistentVolume
metadata:
  annotations:
    pv.kubernetes.io/provisioned-by: everest-csi-provisioner
    everest.io/reclaim-policy: retain-volume-only # (Optional) The PV is
    deleted while the underlying volume is retained.
    name: pv-sfs # PV name.
spec:
  accessModes:
    - ReadWriteMany # Access mode. The value must be ReadWriteMany for SFS.
  capacity:
    storage: 1Gi # SFS volume capacity.
  csi:
    driver: disk.csi.everest.io # Dependent storage driver for the mounting.
    fsType: nfs
    volumeHandle: <your_volume_id> # SFS Capacity-Oriented volume ID.
    volumeAttributes:
      everest.io/share-export-location: <your_location> # Shared path of the
      SFS volume.
      storage.kubernetes.io/csiProvisionerIdentity: everest-csi-provisioner
      persistentVolumeReclaimPolicy: Retain # Reclaim policy.
    storageClassName: csi-nas # Storage class name. csi-nas
    indicates that SFS Capacity-Oriented is used.
    mountOptions: [] # Mount options.
```

**Tabla 8-17** Parámetros clave

Parámetro	Obligatorio	Descripción
everest.io/reclaim-policy: retain-volume-only	No	Opcional. Actualmente, solo se admite <b>retain-volume-only</b> . Este campo solo es válido cuando la versión más reciente es 1.2.9 o posterior y la política de recuperación es <b>Delete</b> . Si la política de reclamación es de <b>Delete</b> y el valor actual es de <b>retain-volume-only</b> el PV asociado se elimina mientras se conserva el volumen de almacenamiento subyacente cuando se elimina un PVC.
volumeHandle	Sí	– Si se utiliza un volumen de SFS Capacity-Oriented, introduzca el ID de volumen. Inicie sesión en la consola, seleccione <b>Service List &gt; Storage &gt; Scalable File Service</b> y seleccione <b>SFS Turbo</b> . En la lista, haga clic en el nombre del sistema de archivos SFS de destino. En la página de detalles, copie el contenido siguiendo <b>ID</b> .

Parámetro	Obligatorio	Descripción
everest.io/share-export-location	Sí	Ruta de acceso compartida del sistema de archivos. – Para un sistema de archivos SFS Capacity-Oriented, inicie sesión en la consola, seleccione <b>Service List &gt; Storage &gt; Scalable File Service</b> y obtenga la ruta de acceso compartida de la columna <b>Mount Address</b> .
mountOptions	Sí	Opciones de montaje. Si no se especifica, se utilizan las siguientes configuraciones de forma predeterminada. Para obtener más información, véase <a href="#">Configuración de las opciones de montaje de volumen de SFS</a> . <pre>mountOptions: - vers=3 - timeo=600 - nolock - hard</pre>
persistentVolumeReclaimPolicy	Sí	Se admite una política de recuperación cuando la versión del clúster es o posterior a 1.19.10 y la versión everest es o posterior a 1.2.9. Las políticas de recuperación <b>Delete</b> y <b>Retain</b> son compatibles. Para obtener más información, véase <a href="#">Política de reclamo de PV</a> . Si varios PV utilizan el mismo volumen de SFS, utilice <b>Retain</b> para evitar la eliminación en cascada de volúmenes subyacentes. <b>Delete:</b> – Si no se especifica <b>everest.io/reclaim-policy</b> , el volumen PV y SFS se eliminan cuando se elimina un PVC. – Si <b>everest.io/reclaim-policy</b> se establece en <b>retain-volume-only set</b> cuando se elimina un PVC, se elimina el PV pero se conservan los recursos de volumen de SFS. <b>Retain:</b> Cuando se elimina un PVC, el PV y los recursos de almacenamiento subyacentes no se eliminan. En su lugar, debe eliminar manualmente estos recursos. Después de eso, el PV está en el estado <b>Released</b> y no puede estar ligado al PVC de nuevo.
storage	Sí	Capacidad solicitada en el PVC, en Gi. Para SFS, este campo solo se utiliza para verificación (no puede estar vacío ni <b>0</b> ). Su valor se fija en <b>1</b> y cualquier valor que establezca no tiene efecto para los sistemas de archivos SFS.

2. Ejecute el siguiente comando para crear un PV:

```
kubectl apply -f pv-sfs.yaml
```

**Paso 3** Cree un PVC.

1. Cree el archivo **pvc-sfs.yaml**.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-sfs
  namespace: default
  annotations:
    volume.beta.kubernetes.io/storage-provisioner: everest-csi-provisioner
spec:
  accessModes:
    - ReadWriteMany # The value must be ReadWriteMany for SFS.
  resources:
    requests:
      storage: 1Gi # SFS volume capacity.
  storageClassName: csi-nas # Storage class name, which must be the same as the
  PV's storage class.
  volumeName: pv-sfs # PV name.
```

**Tabla 8-18** Parámetros clave

Parámetro	Obligatorio	Descripción
storage	Sí	Capacidad solicitada en el PVC, en Gi. El valor debe ser el mismo que el tamaño de almacenamiento del PV existente.
volumeName	Sí	Nombre de PV, que debe ser el mismo que el nombre de PV en <b>1</b> .

2. Ejecute el siguiente comando para crear un PVC:

```
kubectl apply -f pvc-sfs.yaml
```

**Paso 4** Cree una aplicación.

1. Cree un archivo denominado **web-demo.yaml**. En este ejemplo, el volumen de SFS se monta en la ruta **/data**.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web-demo
  namespace: default
spec:
  replicas: 2
  selector:
    matchLabels:
      app: web-demo
  template:
    metadata:
      labels:
        app: web-demo
    spec:
      containers:
        - name: container-1
          image: nginx:latest
          volumeMounts:
```

```

- name: pvc-sfs-volume # Volume name, which must be the same as
the volume name in the volumes field.
  mountPath: /data # Location where the storage volume is mounted.
imagePullSecrets:
- name: default-secret
volumes:
- name: pvc-sfs-volume # Volume name, which can be customized.
  persistentVolumeClaim:
    claimName: pvc-sfs # Name of the created PVC.
    
```

2. Ejecute el siguiente comando para crear una aplicación en la que se monta el volumen de SFS:

```
kubectl apply -f web-demo.yaml
```

Una vez creada la carga de trabajo, puede probar [Verificación de la persistencia y el uso compartido de datos](#).

----Fin

## Verificación de la persistencia y el uso compartido de datos

**Paso 1** Vea las aplicaciones y los archivos desplegados.

1. Ejecute el siguiente comando para ver los pods creados:

```
kubectl get pod | grep web-demo
```

Producto esperado:

```

web-demo-846b489584-mjhm9 1/1 Running 0 46s
web-demo-846b489584-wvv5s 1/1 Running 0 46s
    
```

2. Ejecute los siguientes comandos en secuencia para ver los archivos en la ruta **/data** de los pods:

```

kubectl exec web-demo-846b489584-mjhm9 -- ls /data
kubectl exec web-demo-846b489584-wvv5s -- ls /data
    
```

Si no se devuelve ningún resultado para ambos pods, no existe ningún archivo en la ruta **/data**.

**Paso 2** Ejecute el siguiente comando para crear un archivo llamado **static** en la ruta **/data**:

```
kubectl exec web-demo-846b489584-mjhm9 -- touch /data/static
```

**Paso 3** Ejecute el siguiente comando para ver los archivos en la ruta **/data**:

```
kubectl exec web-demo-846b489584-mjhm9 -- ls /data
```

Producto esperado:

```
static
```

**Paso 4** Verificar la persistencia de los datos.

1. Ejecute el siguiente comando para eliminar el pod llamado **web-demo-846b489584-mjhm9**:

```
kubectl delete pod web-demo-846b489584-mjhm9
```

Producto esperado:

```
pod "web-demo-846b489584-mjhm9" deleted
```

Después de la eliminación, el controlador de Deployment crea automáticamente una réplica.

2. Ejecute el siguiente comando para ver el pod creado:

```
kubectl get pod | grep web-demo
```

El resultado esperado es el siguiente, en el que **web-demo-846b489584-d4d4j** es el pod recién creado:

```

web-demo-846b489584-d4d4j 1/1 Running 0 110s
web-demo-846b489584-wvv5s 1/1 Running 0 7m50s
    
```

- Ejecute el siguiente comando para comprobar si se han modificado los archivos de la ruta **/data** del nuevo pod:

```
kubectl exec web-demo-846b489584-d4d4j -- ls /data
```

Producto esperado:

```
static
```

Si el archivo **static** todavía existe, los datos se pueden almacenar de forma persistente.

### Paso 5 Verifique el uso compartido de datos.

- Ejecute el siguiente comando para ver el pod creado:

```
kubectl get pod | grep web-demo
```

Producto esperado:

```
web-demo-846b489584-d4d4j 1/1 Running 0 7m
web-demo-846b489584-wvv5s 1/1 Running 0 13m
```

- Ejecute el siguiente comando para crear un archivo llamado **share** en la ruta **/data** de cualquier pod: En este ejemplo, seleccione el pod llamado **web-demo-846b489584-d4d4j**.

```
kubectl exec web-demo-846b489584-d4d4j -- touch /data/share
```

Compruebe los archivos en la ruta **/data** del pod.

```
kubectl exec web-demo-846b489584-d4d4j -- ls /data
```

Producto esperado:

```
share
static
```

- Compruebe si el archivo **share** existe en la ruta **/data** de otro pod (**web-demo-846b489584-wvv5s**) también para verificar el uso compartido de datos.

```
kubectl exec web-demo-846b489584-wvv5s -- ls /data
```

Producto esperado:

```
share
static
```

Después de crear un archivo en la ruta **/data** de un pod, si el archivo también se crea en la ruta **/data** de otros pods, los dos pods comparten el mismo volumen.

----Fin

## Operaciones relacionadas

También puede realizar las operaciones que aparecen en [Tabla 8-19](#).

**Tabla 8-19** Operaciones relacionadas

Operación	Descripción	Procedimiento
Creación de un volumen de almacenamiento (PV)	Cree un PV en la consola de CCE.	<ol style="list-style-type: none"> <li data-bbox="833 371 1426 1491">                     Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumes (PVs)</b>. Haga clic en <b>Create Volume</b> en la esquina superior derecha. En el cuadro de diálogo que se muestra, configure los parámetros.                     <ul style="list-style-type: none"> <li data-bbox="874 551 1267 584">● <b>Volume Type:</b> Seleccione <b>SFS</b>.</li> <li data-bbox="874 595 1406 725">● <b>SFS:</b> Haga clic en <b>Select SFS</b>. En la página mostrada, seleccione el sistema de archivos SFS que cumpla con sus requisitos y haga clic en <b>OK</b>.</li> <li data-bbox="874 736 1426 837">● <b>Nombre del PV:</b> Ingrese el nombre del PV. El nombre de PV debe ser único en el mismo clúster.</li> <li data-bbox="874 848 1422 1050">● <b>Access Mode:</b> Los volúmenes de SFS solo admiten <b>ReadWriteMany</b>, lo que indica que un volumen de almacenamiento se puede montar en varios nodos en modo de lectura/escritura. Para obtener más información, véase <a href="#">Modos de acceso al volumen</a>.</li> <li data-bbox="874 1061 1406 1162">● <b>Reclaim Policy:</b> <b>Delete</b> o <b>Retain</b>. Para obtener más información, véase <a href="#">Política de reclamo de PV</a>.</li> </ul> <p data-bbox="908 1173 987 1196"><b>NOTA</b></p> <p data-bbox="932 1200 1406 1312">Si varios PV utilizan el mismo volumen de almacenamiento subyacente, utilice <b>Retain</b> para evitar la eliminación en cascada de volúmenes subyacentes.</p> <ul style="list-style-type: none"> <li data-bbox="874 1323 1422 1491">● <b>Mount Options:</b> Introduzca los pares clave-valor del parámetro de montaje. Para obtener más información, véase <a href="#">Configuración de las opciones de montaje de volumen de SFS</a>.</li> </ul> </li> <li data-bbox="833 1503 1107 1536">2. Haga clic en <b>Create</b>.</li> </ol>
Consulta de eventos	Puede ver los nombres de eventos, los tipos de eventos, el número de ocurrencias, los eventos de Kubernetes, la hora de primera ocurrencia y la hora de última ocurrencia del PVC o PV.	<ol style="list-style-type: none"> <li data-bbox="833 1565 1410 1666">1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumeClaims (PVCs)</b> o <b>PersistentVolumes (PVs)</b>.</li> <li data-bbox="833 1677 1426 1807">2. Haga clic en <b>View Events</b> en la columna <b>Operation</b> del PVC o del PV de destino para ver los eventos generados en una hora (los datos de eventos se conservan durante una hora).</li> </ol>

Operación	Descripción	Procedimiento
Consulta de un archivo YAML	Puede ver, copiar y descargar los archivos YAML de un PVC o PV.	<ol style="list-style-type: none"> <li>1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumeClaims (PVCs)</b> o <b>PersistentVolumes (PVs)</b>.</li> <li>2. Haga clic en <b>View YAML</b> en la columna <b>Operation</b> del PVC o del PV de destino para ver o descargar el YAML.</li> </ol>

### 8.4.3 Uso de un sistema de archivos SFS a través de un PV dinámico

En esta sección se describe cómo utilizar las clases de almacenamiento para crear de forma dinámica PV y PVC e implementar la persistencia y el uso compartido de datos en las cargas de trabajo.

#### Restricciones

- Varios PV pueden utilizar el mismo sistema de archivos SFS o SFS Turbo con las siguientes restricciones:
  - Puede producirse un error si varios PVC/PV que utilizan el mismo sistema de archivos SFS o SFS Turbo subyacente están montados en el mismo pod.
  - El parámetro **persistentVolumeReclaimPolicy** de los PV debe establecerse en **Retain**. De lo contrario, cuando se elimina un PV, se puede eliminar el volumen subyacente asociado. En este caso, otros PV asociados con el volumen subyacente pueden ser anormales.
  - Cuando el volumen subyacente se utiliza repetidamente, se recomienda desplegar **ReadWriteMany** en la capa de aplicación para evitar la sobreescritura y la pérdida de datos.

#### (Consola) Creación automática de un sistema de archivos SFS

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** Cree dinámicamente un PVC y un PV.

1. Elija **Storage** en el panel de navegación y haga clic en la ficha **PersistentVolumeClaims (PVCs)**. Haga clic en **Create PVC** en la esquina superior derecha. En el cuadro de diálogo que se muestra, configure los parámetros de PVC.

Parámetro	Descripción
PVC Type	En este ejemplo, seleccione <b>SFS</b> .
PVC Name	Escriba el nombre de PVC, que debe ser único en el mismo espacio de nombres.

Parámetro	Descripción
Creation Method	<ul style="list-style-type: none"> <li>– Si no hay almacenamiento subyacente disponible, puede seleccionar <b>Dynamically provision</b> para crear un almacenamiento de PVC, PV y subyacente en la consola en modo en cascada.</li> <li>– Si el almacenamiento subyacente está disponible, puede crear un volumen de almacenamiento o utilizar un volumen de almacenamiento existente para crear estáticamente un PVC en función de si se ha creado un PV. Para obtener más información, véase <a href="#">Uso de un sistema de archivos de SFS existente con un PV estático</a>.</li> </ul> <p>En este ejemplo, seleccione <b>Dynamically provision</b>.</p>
Storage Classes	La clase de almacenamiento para los volúmenes SFS es <b>csi-sfs</b> .
Access Mode	Los volúmenes de SFS solo admiten <b>ReadWriteMany</b> , lo que indica que un volumen de almacenamiento se puede montar en varios nodos en modo de lectura/escritura. Para obtener más información, véase <a href="#">Modos de acceso al volumen</a> .

2. Haga clic en **Create** para crear un PVC y un PV.

Puede elegir **Storage** en el panel de navegación y ver el PVC y PV creados en las páginas de fichas **PersistentVolumeClaims (PVCs)** y **PersistentVolumes (PVs)**.

**Paso 3** Cree una aplicación.

1. En el panel de navegación de la izquierda, haga clic en **Workloads**. En el panel derecho, haga clic en la ficha **Deployments**.
2. Haga clic en **Create Workload** en la esquina superior derecha. En la página mostrada, haga clic en **Data Storage** en el área **Container Settings** y haga clic en **Add Volume** para seleccionar **PVC**.

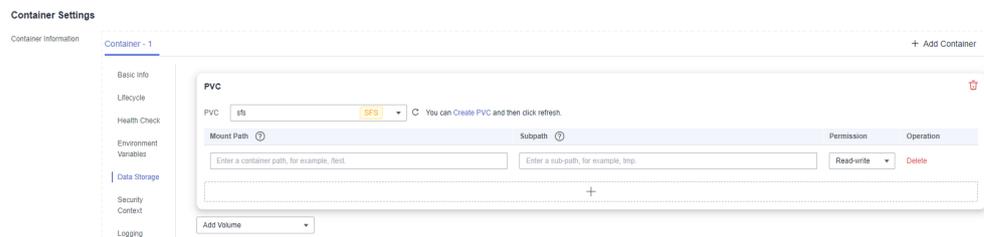
Montar y utilizar volúmenes de almacenamiento, como se muestra en [Tabla 8-20](#). Para obtener más información sobre otros parámetros, consulte [Workloads](#).

**Tabla 8-20** Montaje de un volumen de almacenamiento

Parámetro	Descripción
PVC	Seleccione un volumen de SFS existente.

Parámetro	Descripción
Add Container Path	<p>Configure los parámetros siguientes:</p> <ol style="list-style-type: none"> <li><b>Mount Path:</b> Introduzca una ruta de acceso del contenedor. Por ejemplo, <code>/tmp</code>. Este parámetro indica la ruta de contenedor en la que se montará un volumen de datos. No monte el volumen en un directorio del sistema como <code>/</code> o <code>/var/run</code>; esta acción puede causar errores de contenedor. Se recomienda montar el volumen en un directorio vacío. Si el directorio no está vacío, asegúrese de que no hay archivos que afecten al inicio de contenedor. De lo contrario, se reemplazarán los archivos, lo que provocará errores de inicio de contenedor o errores de creación de carga de trabajo.                     <p><b>AVISO</b></p>                     Cuando el contenedor está montado en un directorio de alto riesgo, se recomienda utilizar una cuenta con los permisos mínimos para iniciar el contenedor; de lo contrario, los archivos de alto riesgo en la máquina host podrían estar dañados.</li> <li><b>Subpath:</b> Introduzca una ruta secundaria, por ejemplo, <code>tmp</code>. Se utiliza una ruta secundaria para montar un volumen local de modo que se utilice el mismo volumen de datos en un solo pod. Si este parámetro se deja en blanco, la ruta raíz se utiliza de forma predeterminada.</li> <li><b>Permission</b> <ul style="list-style-type: none"> <li><b>Read-only:</b> Solo puede leer los datos en los volúmenes montados.</li> <li><b>Read/Write:</b> Puede modificar los volúmenes de datos montados en la ruta de acceso. Los datos recién escritos no se migran si se migra el contenedor, lo que puede causar la pérdida de datos.</li> </ul> </li> </ol> <p>Puede hacer clic en <b>+</b> para agregar varias rutas y subrutas.</p>

En este ejemplo, el disco se monta en la trayectoria `/data` del contenedor. Los datos de contenedor generados en esta ruta se almacenan en el sistema de archivos SFS.



3. Después de completar la configuración, haga clic en **Create**.

Una vez creada la carga de trabajo, puede probar [Verificación de la persistencia y el uso compartido de datos](#).

----Fin

## (kubectl) Creación automática de un sistema de archivos SFS

**Paso 1** Utilice kubectl para conectarse al clúster.

**Paso 2** Utilice **StorageClass** para crear dinámicamente un PVC y un PV.

1. Cree el archivo **pvc-sfs-auto.yaml**.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-sfs-auto
  namespace: default
  annotations: {}
spec:
  accessModes:
    - ReadWriteMany          # The value must be ReadWriteMany for SFS.
  resources:
    requests:
      storage: 1Gi          # SFS volume capacity.
      storageClassName: csi-nas # The storage class type is SFS.
```

**Tabla 8-21** Parámetros clave

Parámetro	Obligatorio	Description
storage	Sí	Capacidad solicitada en el PVC, en Gi. Para SFS, este campo solo se utiliza para verificación (no puede estar vacío ni <b>0</b> ). Su valor se fija en <b>1</b> y cualquier valor que establezca no tiene efecto para los sistemas de archivos SFS.

2. Ejecute el siguiente comando para crear un PVC:

```
kubectl apply -f pvc-sfs-auto.yaml
```

**Paso 3** Cree una aplicación.

1. Cree un archivo denominado **web-demo.yaml**. En este ejemplo, el volumen de SFS se monta en la ruta **/data**.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web-demo
  namespace: default
spec:
  replicas: 2
  selector:
    matchLabels:
      app: web-demo
  template:
    metadata:
      labels:
        app: web-demo
    spec:
      containers:
        - name: container-1
          image: nginx:latest
          volumeMounts:
            - name: pvc-sfs-volume # Volume name, which must be the same as
              # the volume name in the volumes field.
              mountPath: /data # Location where the storage volume is mounted.
          imagePullSecrets:
```

```
- name: default-secret
volumes:
- name: pvc-sfs-volume # Volume name, which can be customized.
  persistentVolumeClaim:
    claimName: pvc-sfs-auto # Name of the created PVC.
```

2. Ejecute el siguiente comando para crear una aplicación en la que se monta el volumen de SFS:

```
kubectl apply -f web-demo.yaml
```

Una vez creada la carga de trabajo, puede probar [Verificación de la persistencia y el uso compartido de datos](#).

----Fin

## Verificación de la persistencia y el uso compartido de datos

**Paso 1** Vea las aplicaciones y los archivos desplegados.

1. Ejecute el siguiente comando para ver los pods creados:

```
kubectl get pod | grep web-demo
```

Producto esperado:

```
web-demo-846b489584-mjhm9 1/1 Running 0 46s
web-demo-846b489584-wvv5s 1/1 Running 0 46s
```

2. Ejecute los siguientes comandos en secuencia para ver los archivos en la ruta **/data** de los pods:

```
kubectl exec web-demo-846b489584-mjhm9 -- ls /data
kubectl exec web-demo-846b489584-wvv5s -- ls /data
```

Si no se devuelve ningún resultado para ambos pods, no existe ningún archivo en la ruta **/data**.

**Paso 2** Ejecute el siguiente comando para crear un archivo llamado **static** en la ruta **/data**:

```
kubectl exec web-demo-846b489584-mjhm9 -- touch /data/static
```

**Paso 3** Ejecute el siguiente comando para ver los archivos en la ruta **/data**:

```
kubectl exec web-demo-846b489584-mjhm9 -- ls /data
```

Producto esperado:

```
static
```

**Paso 4** Verificar la persistencia de los datos.

1. Ejecute el siguiente comando para eliminar el pod llamado **web-demo-846b489584-mjhm9**:

```
kubectl delete pod web-demo-846b489584-mjhm9
```

Producto esperado:

```
pod "web-demo-846b489584-mjhm9" deleted
```

Después de la eliminación, el controlador de Deployment crea automáticamente una réplica.

2. Ejecute el siguiente comando para ver el pod creado:

```
kubectl get pod | grep web-demo
```

El resultado esperado es el siguiente, en el que **web-demo-846b489584-d4d4j** es el pod recién creado:

```
web-demo-846b489584-d4d4j 1/1 Running 0 110s
web-demo-846b489584-wvv5s 1/1 Running 0 7m50s
```

3. Ejecute el siguiente comando para comprobar si se han modificado los archivos de la ruta **/data** del nuevo pod:

```
kubectl exec web-demo-846b489584-d4d4j -- ls /data
```

Producto esperado:

```
static
```

Si el archivo **static** todavía existe, los datos se pueden almacenar de forma persistente.

### Paso 5 Verifique el uso compartido de datos.

1. Ejecute el siguiente comando para ver el pod creado:

```
kubectl get pod | grep web-demo
```

Producto esperado:

```
web-demo-846b489584-d4d4j 1/1 Running 0 7m
web-demo-846b489584-wvv5s 1/1 Running 0 13m
```

2. Ejecute el siguiente comando para crear un archivo llamado **share** en la ruta **/data** de cualquier pod: En este ejemplo, seleccione el pod llamado **web-demo-846b489584-d4d4j**.

```
kubectl exec web-demo-846b489584-d4d4j -- touch /data/share
```

Compruebe los archivos en la ruta **/data** del pod.

```
kubectl exec web-demo-846b489584-d4d4j -- ls /data
```

Producto esperado:

```
share
static
```

3. Compruebe si el archivo **share** existe en la ruta **/data** de otro pod (**web-demo-846b489584-wvv5s**) también para verificar el uso compartido de datos.

```
kubectl exec web-demo-846b489584-wvv5s -- ls /data
```

Producto esperado:

```
share
static
```

Después de crear un archivo en la ruta **/data** de un pod, si el archivo también se crea en la ruta **/data** de otros pods, los dos pods comparten el mismo volumen.

----Fin

## Operaciones relacionadas

También puede realizar las operaciones que aparecen en [Tabla 8-22](#).

**Tabla 8-22** Operaciones relacionadas

Operación	Descripción	Procedimiento
Consulta de eventos	Puede ver los nombres de eventos, los tipos de eventos, el número de ocurrencias, los eventos de Kubernetes, la hora de primera ocurrencia y la hora de última ocurrencia del PVC o PV.	<ol style="list-style-type: none"> <li>1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumeClaims (PVCs)</b> o <b>PersistentVolumes (PVs)</b>.</li> <li>2. Haga clic en <b>View Events</b> en la columna <b>Operation</b> del PVC o del PV de destino para ver los eventos generados en una hora (los datos de eventos se conservan durante una hora).</li> </ol>

Operación	Descripción	Procedimiento
Consulta de un archivo YAML	Puede ver, copiar y descargar los archivos YAML de un PVC o PV.	<ol style="list-style-type: none"> <li>1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumeClaims (PVCs)</b> o <b>PersistentVolumes (PVs)</b>.</li> <li>2. Haga clic en <b>View YAML</b> en la columna <b>Operation</b> del PVC o del PV de destino para ver o descargar el YAML.</li> </ol>

## 8.4.4 Configuración de las opciones de montaje de volumen de SFS

Esta sección describe cómo configurar las opciones de montaje de volumen de SFS. Puede configurar las opciones de montaje en un PV y vincularlo a un PVC. Alternativamente, configure las opciones de montaje en un StorageClass y use el StorageClass para crear un PVC. De esta manera, los PV se pueden crear dinámicamente y heredar opciones de montaje configuradas en el StorageClass de forma predeterminada.

### Requisitos previos

La versión del complemento más antiguo debe ser **1.2.8 o posterior**. El complemento identifica las opciones de montaje y las transfiere a los recursos de almacenamiento subyacentes, que determinan si las opciones especificadas son válidas.

### Restricciones

Las opciones de montaje no se pueden configurar para los contenedores seguros.

### Opciones de montaje de volumen de SFS

El complemento más antiguo de CCE ajusta las opciones descritas en [Tabla 8-23](#) para el montaje de volúmenes de SFS.

**Tabla 8-23** Opciones de montaje de volumen de SFS

Parámetro	Descripción
vers=3	Versión del sistema de archivos. Actualmente, solo se admite NFSv3. Valor: <b>3</b>
nolock	Si se deben bloquear los archivos en el servidor mediante el protocolo NLM. Si se selecciona <b>nolock</b> , el bloqueo es válido para aplicaciones en un host. Para aplicaciones en otro host, el bloqueo no es válido.
timeo=600	Tiempo de espera antes de que el cliente NFS retransmita una solicitud. La unidad es de 0.1 segundos. Valor recomendado: <b>600</b>

Parámetro	Descripción
hard/soft	Modo de montaje. <ul style="list-style-type: none"> <li>● <b>hard</b>: Si el tiempo de espera de la solicitud NFS, el cliente sigue reenviando la solicitud hasta que la solicitud se realiza correctamente.</li> <li>● <b>soft</b>: Si el tiempo de espera de la solicitud NFS, el cliente devuelve un error al programa invocador.</li> </ul> El valor predeterminado es <b>hard</b> .

Puede configurar otras opciones de montaje si es necesario. Para obtener más información, consulte [Montaje de un sistema de archivos NFS en ECS \(Linux\)](#).

## Configuración de opciones de montaje en un PV

Puede utilizar el campo **mountOptions** para establecer las opciones de montaje en un PV. Las opciones que puede configurar en **mountOptions** se enumeran en [Opciones de montaje de volumen de SFS](#).

**Paso 1** Utilice `kubectl` para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).

**Paso 2** Establezca las opciones de montaje en un PV. Por ejemplo:

```

apiVersion: v1
kind: PersistentVolume
metadata:
  annotations:
    pv.kubernetes.io/provisioned-by: everest-csi-provisioner
    everest.io/reclaim-policy: retain-volume-only # (Optional) The PV is
    deleted while the underlying volume is retained.
  name: pv-sfs
spec:
  accessModes:
    - ReadWriteMany # Access mode. The value must be ReadWriteMany for SFS.
  capacity:
    storage: 1Gi # SFS volume capacity.
  csi:
    driver: disk.csi.everest.io # Dependent storage driver for the mounting.
    fsType: nfs
    volumeHandle: <your_volume_id> # ID of the SFS Capacity-Oriented volume.
    volumeAttributes:
      everest.io/share-export-location: <your_location> # Shared path of the SFS
    volume.
    storage.kubernetes.io/csiProvisionerIdentity: everest-csi-provisioner
  persistentVolumeReclaimPolicy: Retain # Reclaim policy.
  storageClassName: csi-nas # Storage class name.
  mountOptions: # Mount options.
    - vers=3
    - noLock
    - timeo=600
    - hard
    
```

**Paso 3** Después de crear un PV, puede crear un PVC y vincularlo al PV y, a continuación, montar el PV en el contenedor en la carga de trabajo. Para obtener más información, véase [Uso de un sistema de archivos de SFS existente con un PV estático](#).

**Paso 4** Compruebe si las opciones de montaje tienen efecto.

En este ejemplo, el PVC se monta en la carga de trabajo que utiliza la imagen **nginx:latest**. Puede ejecutar el comando **mount -l** para comprobar si las opciones de montaje tienen efecto.

1. Vea el pod en el que se ha montado el volumen de SFS. En este ejemplo, el nombre de la carga de trabajo es **web-sfs**.

```
kubectl get pod | grep web-sfs
```

Salida del comando:

```
web-sfs-*** 1/1 Running 0 23m
```

2. Ejecute el siguiente comando para comprobar las opciones de montaje (**web-sfs-\*\*\*** es un pod de ejemplo):

```
kubectl exec -it web-sfs-*** -- mount -l | grep nfs
```

Si la información de montaje en la salida del comando es consistente con las opciones de montaje configuradas, las opciones de montaje se establecen correctamente.

```
<Your shared path> on /data type nfs
(rw,relatime,vers=3,rsize=1048576,wsz=1048576,namlen=255,hard,nolock,noresvp
ort,proto=tcp,timeo=600,retrans=2,sec=sys,mountaddr=*.**.**.**,mountvers=3,m
ountport=2050,mountproto=tcp,local_lock=all,addr=*.**.**.**)

```

----Fin

## Configuración de las opciones de montaje en un StorageClass

Puede utilizar el campo **mountOptions** para establecer las opciones de montaje en un StorageClass. Las opciones que puede configurar en **mountOptions** se enumeran en [Opciones de montaje de volumen de SFS](#).

- Paso 1** Utilice kubectl para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).

- Paso 2** Cree un StorageClass personalizado. Por ejemplo:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-sfs-mount-option
provisioner: everest-csi-provisioner
parameters:
  csi.storage.k8s.io/csi-driver-name: nas.csi.everest.io
  csi.storage.k8s.io/fstype: nfs
  everest.io/share-access-to: <your_vpc_id> # VPC ID of the cluster.
  reclaimPolicy: Delete
  volumeBindingMode: Immediate
mountOptions: # Mount options.
- vers=3
- nolock
- timeo=600
- hard
```

- Paso 3** Una vez configurado el StorageClass, puede usarlo para crear un PVC. De forma predeterminada, los PV creados dinámicamente heredan las opciones de montaje establecidas en el StorageClass. Para obtener más información, véase [Uso de un sistema de archivos SFS a través de un PV dinámico](#).

- Paso 4** Compruebe si las opciones de montaje tienen efecto.

En este ejemplo, el PVC se monta en la carga de trabajo que utiliza la imagen **nginx:latest**. Puede ejecutar el comando **mount -l** para comprobar si las opciones de montaje tienen efecto.

1. Vea el pod en el que se ha montado el volumen de SFS. En este ejemplo, el nombre de la carga de trabajo es **web-sfs**.

```
kubectl get pod | grep web-sfs
```

Salida del comando:

```
web-sfs-*** 1/1 Running 0 23m
```

2. Ejecute el siguiente comando para comprobar las opciones de montaje (**web-sfs-\*\*\*** es un pod de ejemplo):

```
kubectl exec -it web-sfs-*** -- mount -l | grep nfs
```

Si la información de montaje en la salida del comando es consistente con las opciones de montaje configuradas, las opciones de montaje se establecen correctamente.

```
<Your shared path> on /data type nfs
(rw,relatime,vers=3,rsize=1048576,wsiz=1048576,namlen=255,hard,nolock,norexport,proto=tcp,timeo=600,retrans=2,sec=sys,mountaddr=*. *. *. *. *,mountvers=3,mountport=2050,mountproto=tcp,local_lock=all,addr=*. *. *. *. *)
```

----Fin

## 8.5 Sistemas de archivos SFS Turbo

### 8.5.1 Descripción general

#### Presentación

CCE le permite montar los volúmenes de almacenamiento creados por sistemas de archivos SFS Turbo en una ruta de un contenedor para cumplir con los requisitos de persistencia de datos. Los sistemas de archivos SFS Turbo son rápidos, bajo demanda y escalables, que son adecuados para escenarios con una gran cantidad de archivos pequeños, como DevOps y aplicaciones de oficina empresarial.

Ampliable a 320 TB, SFS Turbo proporciona un almacenamiento de archivos compartido totalmente alojado, que es altamente disponible y estable, para admitir archivos y aplicaciones pequeños que requieren baja latencia y alta IOPS.

- **Standard file protocols:** Puede montar sistemas de archivos como volúmenes en servidores, lo mismo que usar directorios locales.
- **Data sharing:** El mismo sistema de archivos se puede montar en varios servidores, para que los datos se puedan compartir.
- **Private network:** Los usuarios solo pueden acceder a los datos en redes privadas de centros de datos.
- **Data isolation:** El servicio de almacenamiento en la nube proporciona almacenamiento exclusivo de archivos en la nube, que ofrece aislamiento de datos y garantiza el rendimiento de IOPS.
- **Use cases:** Deployments/StatefulSets en el modo ReadWriteMany, DaemonSets y trabajos creados para sitios web de alto tráfico, almacenamiento de registros, DevOps y aplicaciones de OA empresariales

#### Rendimiento de SFS Turbo

Para obtener más información sobre los parámetros de rendimiento de SFS Turbo, consulte [Tipos de sistema de archivos](#).

#### Escenario

SFS Turbo admite los siguientes modos de montaje:

- **Uso de un sistema de archivos de SFS Turbo existente con un PV estático:** modo de creación estática, en el que se utiliza un volumen de SFS existente para crear un PV y, a continuación, montar el almacenamiento en la carga de trabajo con un PVC.
- **Creación y montaje dinámico de subdirectorios de un sistema de archivos de SFS Turbo:** SFS Turbo le permite crear subdirectorios dinámicamente y montarlos en contenedores para que SFS Turbo se pueda compartir y la capacidad de almacenamiento SFS Turbo se pueda utilizar de manera más económica y adecuada.

## Restricciones

- El sistema de archivos de SFS y el clúster deben estar en la misma VPC.
- Varios PV pueden utilizar el mismo sistema de archivos SFS o SFS Turbo con las siguientes restricciones:
  - Puede producirse un error si varios PVC/PV que utilizan el mismo sistema de archivos SFS o SFS Turbo subyacente están montados en el mismo pod.
  - El parámetro **persistentVolumeReclaimPolicy** de los PV debe establecerse en **Retain**. De lo contrario, cuando se elimina un PV, se puede eliminar el volumen subyacente asociado. En este caso, otros PV asociados con el volumen subyacente pueden ser anormales.
  - Cuando el volumen subyacente se utiliza repetidamente, se recomienda desplegar **ReadWriteMany** en la capa de aplicación para evitar la sobrescritura y la pérdida de datos.

## Facturación

SFS Turbo no admite la creación dinámica. Solo se pueden montar los volúmenes de SFS Turbo creados. Puede seleccionar el modo de facturación de pago por uso o el paquete anual/mensual según sea necesario. Para obtener más información sobre los precios de SFS Turbo, consulte [Facturación](#).

## 8.5.2 Uso de un sistema de archivos de SFS Turbo existente con un PV estático

SFS Turbo es un sistema de archivos compartido con alta disponibilidad y durabilidad. Es adecuado para aplicaciones que contienen archivos pequeños masivos y requieren baja latencia y alta IOPS. Esta sección describe cómo utilizar un sistema de archivos SFS Turbo existente para crear estáticamente los PV y PVC e implementar la persistencia de datos y el uso compartido en las cargas de trabajo.

### Requisitos previos

- Ha creado un clúster e instalado el complemento de CSI ([everest](#)) en el clúster.
- Si desea crear un clúster mediante comandos, utilice `kubectrl` para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectrl](#).
- Ha creado un sistema de archivos de SFS Turbo disponible, y el sistema de archivos SFS Turbo y el clúster están en la misma VPC.

## Restricciones

- El sistema de archivos de SFS y el clúster deben estar en la misma VPC.

- Varios PV pueden utilizar el mismo sistema de archivos SFS o SFS Turbo con las siguientes restricciones:
  - Puede producirse un error si varios PVC/PV que utilizan el mismo sistema de archivos SFS o SFS Turbo subyacente están montados en el mismo pod.
  - El parámetro **persistentVolumeReclaimPolicy** de los PV debe establecerse en **Retain**. De lo contrario, cuando se elimina un PV, se puede eliminar el volumen subyacente asociado. En este caso, otros PV asociados con el volumen subyacente pueden ser anormales.
  - Cuando el volumen subyacente se utiliza repetidamente, se recomienda desplegar ReadWriteMany en la capa de aplicación para evitar la sobreescritura y la pérdida de datos.
- Para el almacenamiento de SFS Turbo, los recursos de SFS Turbo anuales/mensuales no se recuperarán cuando se elimine el clúster o el PVC. Necesita recuperar los recursos en la consola de SFS Turbo.

## Uso de un sistema de archivos SFS Turbo existente en la consola

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** Cree un PV y PVC estáticamente.

1. Elija **Storage** en el panel de navegación y haga clic en la ficha **PersistentVolumeClaims (PVCs)**. Haga clic en **Create PVC** en la esquina superior derecha. En el cuadro de diálogo que se muestra, configure los parámetros de PVC.

Parámetro	Descripción
PVC Type	En esta sección, seleccione <b>SFS Turbo</b> .
PVC Name	Escriba el nombre de PVC, que debe ser único en el mismo espacio de nombres.
Creation Method	Puede crear un volumen de almacenamiento o utilizar un volumen de almacenamiento existente para crear estáticamente un PVC en función de si se ha creado un PV.  En este ejemplo, seleccione <b>Create new</b> para crear un PV y un PVC al mismo tiempo en la consola.
PV <sup>a</sup>	Seleccione un volumen de PV existente en el clúster. Es necesario crear un PV por adelantado. Para obtener más información, consulte "Creación de un volumen de almacenamiento (PV)" de <a href="#">Operaciones relacionadas</a> .  No es necesario especificar este parámetro en este ejemplo.
SFS Turbo <sup>b</sup>	Haga clic en <b>Select SFS Turbo</b> . En la página que se muestra, seleccione el sistema de archivos SFS Turbo que cumpla con sus requisitos y haga clic en <b>OK</b> .
PV Name <sup>b</sup>	Introduzca el nombre de PV, que debe ser único en el mismo clúster.

Parámetro	Descripción
Access Mode <sup>b</sup>	Los volúmenes de SFS Turbo solo admiten <b>ReadWriteMany</b> , lo que indica que un volumen de almacenamiento se puede montar en varios nodos en modo de lectura/escritura. Para obtener más información, véase <a href="#">Modos de acceso al volumen</a> .
Reclaim Policy <sup>b</sup>	Solo se admite <b>Retain</b> , lo que indica que el PV no se elimina cuando se elimina el PVC. Para obtener más información, véase <a href="#">Política de reclamo de PV</a> .
Mount Options <sup>b</sup>	Introduzca los pares de clave-valor del parámetro de montaje. Para obtener más información, véase <a href="#">Configuración de las opciones de montaje de SFS Turbo</a> .

 **NOTA**

- a: El parámetro está disponible cuando **Creation Method** se establece en **Use existing**.
- b: El parámetro está disponible cuando **Creation Method** se establece en **Create new**.

2. Haga clic en **Create** para crear un PVC y un PV.

Puede elegir **Storage** en el panel de navegación y ver el PVC y PV creados en las páginas de fichas **PersistentVolumeClaims (PVCs)** y **PersistentVolumes (PVs)**.

**Paso 3** Cree una aplicación.

1. En el panel de navegación de la izquierda, haga clic en **Workloads**. En el panel derecho, haga clic en la ficha **Deployments**.
2. Haga clic en **Create Workload** en la esquina superior derecha. En la página mostrada, haga clic en **Data Storage** en el área **Container Settings** y haga clic en **Add Volume** para seleccionar **PVC**.

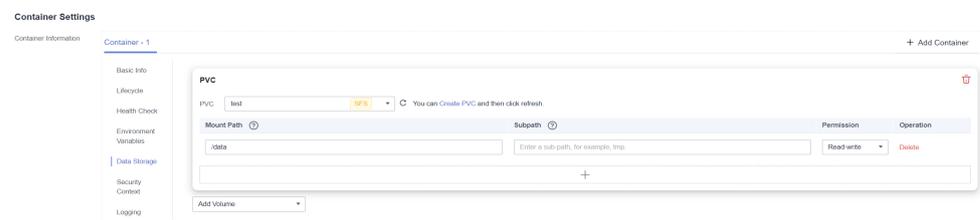
Montar y utilizar volúmenes de almacenamiento, como se muestra en [Tabla 8-24](#). Para obtener más información sobre otros parámetros, consulte [Workloads](#).

**Tabla 8-24** Montaje de un volumen de almacenamiento

Parámetro	Descripción
PVC	Seleccione un volumen de SFS Turbo existente.

Parámetro	Descripción
Add Container Path	<p>Configure los parámetros siguientes:</p> <ol style="list-style-type: none"> <li><b>Mount Path:</b> Introduzca una ruta de acceso del contenedor. Por ejemplo, <code>/tmp</code>. Este parámetro indica la ruta de contenedor en la que se montará un volumen de datos. No monte el volumen en un directorio del sistema como <code>/</code> o <code>/var/run</code>; esta acción puede causar errores de contenedor. Se recomienda montar el volumen en un directorio vacío. Si el directorio no está vacío, asegúrese de que no hay archivos que afecten al inicio de contenedor. De lo contrario, se reemplazarán los archivos, lo que provocará errores de inicio de contenedor o errores de creación de carga de trabajo.                     <p><b>AVISO</b></p>                     Cuando el contenedor está montado en un directorio de alto riesgo, se recomienda utilizar una cuenta con los permisos mínimos para iniciar el contenedor; de lo contrario, los archivos de alto riesgo en la máquina host podrían estar dañados.</li> <li><b>Subpath:</b> Introduzca una ruta secundaria, por ejemplo, <code>tmp</code>. Se utiliza una ruta secundaria para montar un volumen local de modo que se utilice el mismo volumen de datos en un solo pod. Si este parámetro se deja en blanco, la ruta raíz se utiliza de forma predeterminada.</li> <li><b>Permission</b> <ul style="list-style-type: none"> <li><b>Read-only:</b> Solo puede leer los datos en los volúmenes montados.</li> <li><b>Read/Write:</b> Puede modificar los volúmenes de datos montados en la ruta de acceso. Los datos recién escritos no se migran si se migra el contenedor, lo que puede causar la pérdida de datos.</li> </ul> </li> </ol> <p>Puede hacer clic en <b>+</b> para agregar varias rutas y subrutas.</p>

En este ejemplo, el disco se monta en la trayectoria `/data` del contenedor. Los datos de contenedor generados en esta ruta se almacenan en el sistema de archivos SFS Turbo.



- Después de completar la configuración, haga clic en **Create**. Una vez creada la carga de trabajo, puede probar [Verificación de la persistencia y el uso compartido de datos](#).

----Fin

## (kubectl) Uso de un sistema de archivos SFS existente

**Paso 1** Utilice kubectl para conectarse al clúster.

**Paso 2** Cree un PV.

1. Cree el archivo `pv-sfsturbo.yaml`.

```

apiVersion: v1
kind: PersistentVolume
metadata:
  annotations:
    pv.kubernetes.io/provisioned-by: everest-csi-provisioner
  name: pv-sfsturbo # PV name.
spec:
  accessModes:
    - ReadWriteMany # Access mode. The value must be ReadWriteMany for SFS
    Turbo.
  capacity:
    storage: 500Gi # SFS Turbo volume capacity.
  csi:
    driver: sfsturbo.csi.everest.io # Dependent storage driver for the
    mounting.
    fsType: nfs
    volumeHandle: <your_volume_id> # SFS Turbo volume ID.
    volumeAttributes:
      everest.io/share-export-location: <your_location> # Shared path of
    the SFS Turbo volume.
      everest.io/enterprise-project-id: <your_project_id> # Project ID of
    the SFS Turbo volume.
      storage.kubernetes.io/csiProvisionerIdentity: everest-csi-provisioner
    persistentVolumeReclaimPolicy: Retain # Reclaim policy.
    storageClassName: csi-sfsturbo # Storage class name of the SFS
    Turbo file system.
  mountOptions: [] # Mount options.
    
```

**Tabla 8-25** Parámetros clave

Parámetro	Obligatorio	Descripción
volumeHandle	Sí	ID de volumen de SFS Turbo. Cómo obtenerlo: Inicie sesión en la consola, elija <b>Service List &gt; Storage &gt; Scalable File Service</b> y seleccione <b>SFS Turbo</b> . En la lista, haga clic en el nombre del volumen de SFS Turbo de destino. En la página de detalles, copie el contenido siguiendo <b>ID</b> .
everest.io/share-export-location	Sí	Ruta de acceso compartida del volumen de SFS Turbo. Inicie sesión en la consola, seleccione <b>Service List &gt; Storage &gt; Scalable File Service</b> y seleccione <b>SFS Turbo</b> . Puede obtener la ruta de acceso compartida del sistema de archivos desde la columna <b>Mount Address</b> .

Parámetro	Obligatorio	Descripción
everest.io/enterprise-project-id	No	ID de proyecto del volumen de SFS Turbo. Cómo obtenerlo: En la consola de SFS, haga clic en <b>SFS Turbo</b> en el panel de navegación izquierdo. Haga clic en el nombre del sistema de archivos de SFS Turbo para interconectar. En la ficha <b>Basic Info</b> , busque y haga clic en el proyecto de empresa para ir a la consola y copie el ID.
mountOptions	No	Opciones de montaje. Si no se especifica, se utilizan las siguientes configuraciones de forma predeterminada. Para obtener más información, véase <a href="#">Configuración de las opciones de montaje de SFS Turbo</a> . <pre>mountOptions: - vers=3 - timeo=600 - nolock - hard</pre>
persistentVolumeReclaimPolicy	Sí	Se admite una política de recuperación cuando la versión del clúster es o posterior a 1.19.10 y la versión everest es o posterior a 1.2.9. Solo se admite la política de recuperación <b>Retain</b> . Para obtener más información, véase <a href="#">Política de reclamo de PV</a> . <b>Retain</b> : Cuando se elimina un PVC, el PV y los recursos de almacenamiento subyacentes no se eliminan. En su lugar, debe eliminar manualmente estos recursos. Después de eso, el PV está en el estado <b>Released</b> y no puede estar ligado al PVC de nuevo.
storage	Sí	Capacidad solicitada en el PVC, en Gi.
storageClassName	Sí	El nombre de clase de almacenamiento de los volúmenes de SFS Turbo es <b>csi-sfsturbo</b> .

- Ejecute el siguiente comando para crear un PV:

```
kubectl apply -f pv-sfsturbo.yaml
```

### Paso 3 Cree un PVC.

- Cree el archivo **pvc-sfsturbo.yaml**.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-sfsturbo
  namespace: default
  annotations:
    volume.beta.kubernetes.io/storage-provisioner: everest-csi-provisioner
    everest.io/enterprise-project-id: <your_project_id> # Project ID of the
SFS Turbo volume.
spec:
```

```

accessModes:
- ReadWriteMany          # The value must be ReadWriteMany for SFS
Turbo.
resources:
  requests:
    storage: 500Gi        # SFS Turbo volume capacity.
    storageClassName: csi-sfsturbo # Storage class of the SFS Turbo
    volume, which must be the same as that of the PV.
    volumeName: pv-sfsturbo # PV name.
    
```

**Tabla 8-26** Parámetros clave

Parámetro	Obligatorio	Descripción
everest.io/enterprise-project-id	No	ID de proyecto del volumen de SFS Turbo. Cómo obtenerlo: En la consola de SFS, haga clic en <b>SFS Turbo</b> en el panel de navegación izquierdo. Haga clic en el nombre del sistema de archivos de SFS Turbo para interconectar. En la ficha <b>Basic Info</b> , busque y haga clic en el proyecto de empresa para ir a la consola y copie el ID.
storage	Sí	Capacidad solicitada en el PVC, en Gi. El valor debe ser el mismo que el tamaño de almacenamiento del PV existente.
storageClassName	Sí	Nombre de la clase de almacenamiento, que debe ser el mismo que la clase de almacenamiento del PV en <a href="#">1</a> . El nombre de clase de almacenamiento de los volúmenes de SFS Turbo es <b>csi-sfsturbo</b> .
volumeName	Sí	Nombre de PV, que debe ser el mismo que el nombre de PV en <a href="#">1</a> .

- Ejecute el siguiente comando para crear un PVC:

```
kubectl apply -f pvc-sfsturbo.yaml
```

**Paso 4** Cree una aplicación.

- Cree un archivo denominado **web-demo.yaml**. En este ejemplo, el volumen de SFS Turbo se monta en el camino **/data**.

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: web-demo
  namespace: default
spec:
  replicas: 2
  selector:
    matchLabels:
      app: web-demo
  template:
    metadata:
      labels:
        app: web-demo
    spec:
      containers:
    
```

```
- name: container-1
  image: nginx:latest
  volumeMounts:
    - name: pvc-sfsturbo-volume      #Volume name, which must be the same
      as the volume name in the volumes field.
      mountPath: /data #Location where the storage volume is mounted.
  imagePullSecrets:
    - name: default-secret
  volumes:
    - name: pvc-sfsturbo-volume      #Volume name, which can be customized.
      persistentVolumeClaim:
        claimName: pvc-sfsturbo      #Name of the created PVC.
```

2. Ejecute el siguiente comando para crear una aplicación en la que esté montado el volumen de SFS Turbo:

```
kubectl apply -f web-demo.yaml
```

Una vez creada la carga de trabajo, puede probar [Verificación de la persistencia y el uso compartido de datos](#).

----Fin

## Verificación de la persistencia y el uso compartido de datos

**Paso 1** Vea las aplicaciones y los archivos desplegados.

1. Ejecute el siguiente comando para ver los pods creados:

```
kubectl get pod | grep web-demo
```

Producto esperado:

```
web-demo-846b489584-mjhm9    1/1      Running    0          46s
web-demo-846b489584-wvv5s    1/1      Running    0          46s
```

2. Ejecute los siguientes comandos en secuencia para ver los archivos en la ruta **/data** de los pods:

```
kubectl exec web-demo-846b489584-mjhm9 -- ls /data
kubectl exec web-demo-846b489584-wvv5s -- ls /data
```

Si no se devuelve ningún resultado para ambos pods, no existe ningún archivo en la ruta **/data**.

**Paso 2** Ejecute el siguiente comando para crear un archivo llamado **static** en la ruta **/data**:

```
kubectl exec web-demo-846b489584-mjhm9 -- touch /data/static
```

**Paso 3** Ejecute el siguiente comando para ver los archivos en la ruta **/data**:

```
kubectl exec web-demo-846b489584-mjhm9 -- ls /data
```

Producto esperado:

```
static
```

**Paso 4** Verificar la persistencia de los datos.

1. Ejecute el siguiente comando para eliminar el pod llamado **web-demo-846b489584-mjhm9**:

```
kubectl delete pod web-demo-846b489584-mjhm9
```

Producto esperado:

```
pod "web-demo-846b489584-mjhm9" deleted
```

Después de la eliminación, el controlador de Deployment crea automáticamente una réplica.

2. Ejecute el siguiente comando para ver el pod creado:

```
kubectl get pod | grep web-demo
```

El resultado esperado es el siguiente, en el que **web-demo-846b489584-d4d4j** es el pod recién creado:

```
web-demo-846b489584-d4d4j 1/1 Running 0 110s
web-demo-846b489584-wvv5s 1/1 Running 0 7m50s
```

- Ejecute el siguiente comando para comprobar si se han modificado los archivos de la ruta **/data** del nuevo pod:

```
kubectl exec web-demo-846b489584-d4d4j -- ls /data
```

Producto esperado:

```
static
```

Si el archivo **static** todavía existe, los datos se pueden almacenar de forma persistente.

### Paso 5 Verifique el uso compartido de datos.

- Ejecute el siguiente comando para ver el pod creado:

```
kubectl get pod | grep web-demo
```

Producto esperado:

```
web-demo-846b489584-d4d4j 1/1 Running 0 7m
web-demo-846b489584-wvv5s 1/1 Running 0 13m
```

- Ejecute el siguiente comando para crear un archivo llamado **share** en la ruta **/data** de cualquier pod: En este ejemplo, seleccione el pod llamado **web-demo-846b489584-d4d4j**.

```
kubectl exec web-demo-846b489584-d4d4j -- touch /data/share
```

Compruebe los archivos en la ruta **/data** del pod.

```
kubectl exec web-demo-846b489584-d4d4j -- ls /data
```

Producto esperado:

```
share
static
```

- Compruebe si el archivo **share** existe en la ruta **/data** de otro pod (**web-demo-846b489584-wvv5s**) también para verificar el uso compartido de datos.

```
kubectl exec web-demo-846b489584-wvv5s -- ls /data
```

Producto esperado:

```
share
static
```

Después de crear un archivo en la ruta **/data** de un pod, si el archivo también se crea en la ruta **/data** de otros pods, los dos pods comparten el mismo volumen.

----Fin

## Operaciones relacionadas

También puede realizar las operaciones que aparecen en [Tabla 8-27](#).

**Tabla 8-27** Operaciones relacionadas

Operación	Descripción	Procedimiento
Creación de un volumen de almacenamiento (PV)	Cree un PV en la consola de CCE.	<ol style="list-style-type: none"> <li>1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumes (PVs)</b>. Haga clic en <b>Create Volume</b> en la esquina superior derecha. En el cuadro de diálogo que se muestra, configure los parámetros.                     <ul style="list-style-type: none"> <li>● <b>Volume Type:</b> Seleccione <b>SFS Turbo</b>.</li> <li>● <b>SFS Turbo:</b> Haga clic en <b>Select SFS Turbo</b>. En la página que se muestra, seleccione el volumen de SFS Turbo que cumpla con los requisitos y haga clic en <b>OK</b>.</li> <li>● <b>PV Name:</b> Introduzca el nombre de PV, que debe ser único en el mismo clúster.</li> <li>● <b>Access Mode:</b> Los volúmenes de SFS solo admiten <b>ReadWriteMany</b>, lo que indica que un volumen de almacenamiento se puede montar en varios nodos en modo de lectura/escritura. Para obtener más información, véase <a href="#">Modos de acceso al volumen</a>.</li> <li>● <b>Reclaim Policy:</b> Solo se admite el uso de <b>Retain</b>. Para obtener más información, véase <a href="#">Política de reclamo de PV</a>.</li> <li>● <b>Mount Options:</b> Introduzca los pares clave-valor del parámetro de montaje. Para obtener más información, véase <a href="#">Configuración de las opciones de montaje de SFS Turbo</a>.</li> </ul> </li> <li>2. Haga clic en <b>Create</b>.</li> </ol>
Ampliación de la capacidad de un volumen de SFS Turbo	Amplíe rápidamente la capacidad de un volumen de SFS Turbo montado en la consola de CCE.	<ol style="list-style-type: none"> <li>1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumeClaims (PVCs)</b>. Haga clic en <b>More</b> en la columna <b>Operation</b> del PVC de destino y seleccione <b>Scale-out</b>.</li> <li>2. Ingrese la capacidad que desea agregar y haga clic en <b>OK</b>.</li> </ol>
Eventos	Puede ver los nombres de eventos, los tipos de eventos, el número de ocurrencias, los eventos de Kubernetes, la hora de primera ocurrencia y la hora de última ocurrencia del PVC o PV.	<ol style="list-style-type: none"> <li>1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumeClaims (PVCs)</b> o <b>PersistentVolumes (PVs)</b>.</li> <li>2. Haga clic en <b>View Events</b> en la columna <b>Operation</b> del PVC o del PV de destino para ver los eventos generados en una hora (los datos de eventos se conservan durante una hora).</li> </ol>

Operación	Descripción	Procedimiento
Consulta de un archivo YAML	Puede ver, copiar y descargar los archivos YAML de un PVC o PV.	<ol style="list-style-type: none"> <li>1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumeClaims (PVCs)</b> o <b>PersistentVolumes (PVs)</b>.</li> <li>2. Haga clic en <b>View YAML</b> en la columna <b>Operation</b> del PVC o del PV de destino para ver o descargar el YAML.</li> </ol>

### 8.5.3 Configuración de las opciones de montaje de SFS Turbo

Esta sección describe cómo configurar las opciones de montaje de SFS Turbo. Para SFS Turbo, solo puede establecer las opciones de montaje en un PV y vincularlo creando un PVC.

#### Requisitos previos

La versión del complemento más antiguo debe ser **1.2.8 o posterior**. El complemento identifica las opciones de montaje y las transfiere a los recursos de almacenamiento subyacentes, que determinan si las opciones especificadas son válidas.

#### Restricciones

Las opciones de montaje no se pueden configurar para el contenedor de Kata.

#### Opciones de montaje de SFS Turbo

El complemento más antiguo de CCE preajusta las opciones descritas en [Tabla 8-28](#) para el montaje de volúmenes de SFS Turbo.

**Tabla 8-28** Opciones de montaje de SFS Turbo

Parámetro	Descripción
vers=3	Versión del sistema de archivos. Actualmente, solo se admite NFSv3. Valor: <b>3</b>
nolock	Si se deben bloquear los archivos en el servidor mediante el protocolo NLM. Si se selecciona <b>nolock</b> , el bloqueo es válido para aplicaciones en un host. Para aplicaciones en otro host, el bloqueo no es válido.
timeo=600	Tiempo de espera antes de que el cliente NFS retransmita una solicitud. La unidad es de 0.1 segundos. Valor recomendado: <b>600</b>

Parámetro	Descripción
hard/soft	<p>Modo de montaje.</p> <ul style="list-style-type: none"> <li>● <b>hard</b>: Si el tiempo de espera de la solicitud NFS, el cliente sigue reenviando la solicitud hasta que la solicitud se realiza correctamente.</li> <li>● <b>soft</b>: Si el tiempo de espera de la solicitud NFS, el cliente devuelve un error al programa invocador.</li> </ul> <p>El valor predeterminado es <b>hard</b>.</p>

Puede configurar otras opciones de montaje si es necesario. Para obtener más información, consulte [Montaje de un sistema de archivos NFS en ECS \(Linux\)](#).

## Configuración de opciones de montaje en un PV

Puede utilizar el campo **mountOptions** para configurar las opciones de montaje en un PV. Las opciones que puede configurar en **mountOptions** se enumeran en [Opciones de montaje de SFS Turbo](#).

**Paso 1** Utilice kubectl para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).

**Paso 2** Establezca las opciones de montaje en un PV. Por ejemplo:

```

apiVersion: v1
kind: PersistentVolume
metadata:
  annotations:
    pv.kubernetes.io/provisioned-by: everest-csi-provisioner
  name: pv-sfsturbo # PV name.
spec:
  accessModes:
    - ReadWriteMany # Access mode. The value must be ReadWriteMany for SFS Turbo.
  capacity:
    storage: 500Gi # SFS Turbo volume capacity.
  csi:
    driver: sfsturbo.csi.everest.io # Dependent storage driver for the mounting.
    fsType: nfs
    volumeHandle: {your_volume_id} # SFS Turbo volume ID
    volumeAttributes:
      everest.io/share-export-location: {your_location} # Shared path of the SFS Turbo volume.
      everest.io/enterprise-project-id: {your_project_id} # Project ID of the SFS Turbo volume.
    storage.kubernetes.io/csiProvisionerIdentity: everest-csi-provisioner
  persistentVolumeReclaimPolicy: Retain # Reclaim policy.
  storageClassName: csi-sfsturbo # SFS Turbo storage class name.
  mountOptions: # Mount options.
    - vers=3
    - nolock
    - timeo=600
    - hard
    
```

**Paso 3** Después de crear un PV, puede crear un PVC y vincularlo al PV y, a continuación, montar el PV en el contenedor en la carga de trabajo. Para obtener más información, véase [Uso de un sistema de archivos de SFS Turbo existente con un PV estático](#).

**Paso 4** Compruebe si las opciones de montaje tienen efecto.

En este ejemplo, el PVC se monta en la carga de trabajo que utiliza la imagen **nginx:latest**. Puede ejecutar el comando **mount -l** para comprobar si las opciones de montaje tienen efecto.

1. Vea el pod en el que se ha montado el volumen SFS Turbo. En este ejemplo, el nombre de la carga de trabajo es **web-sfsturbo**.

```
kubectl get pod | grep web-sfsturbo
```

Salida del comando:

```
web-sfsturbo-*** 1/1 Running 0 23m
```

2. Ejecute el siguiente comando para comprobar las opciones de montaje (**web-sfsturbo-\*\*\*** es un pod de ejemplo):

```
kubectl exec -it web-sfsturbo-*** -- mount -l | grep nfs
```

Si la información de montaje en la salida del comando es consistente con las opciones de montaje configuradas, se han configurado las opciones de montaje.

```
<Your mount path> on /data type nfs  
(rw,relatime,vers=3,rsize=1048576,wsiz=1048576,namlen=255,hard,nolock,noresvport,proto=tcp,timeo=600,retrans=2,sec=sys,mountaddr=*. *. *. *. *,mountvers=3,mountport=20048,mountproto=tcp,local_lock=all,addr=*. *. *. *)
```

----Fin

## 8.5.4 Creación y montaje dinámico de subdirectorios de un sistema de archivos de SFS Turbo

### Antecedentes

La capacidad mínima de un sistema de archivos de SFS Turbo es de 500 GiB, y el sistema de archivos de SFS Turbo no se puede facturar por uso. De forma predeterminada, el directorio raíz de un sistema de archivos de SFS Turbo está montado en un contenedor que, en la mayoría de los casos, no requiere una capacidad tan grande.

El complemento everest le permite crear de forma dinámica subdirectorios en un sistema de archivos de SFS Turbo y montar estos subdirectorios en contenedores. De esta manera, un sistema de archivos de SFS Turbo puede ser compartido por múltiples contenedores para aumentar la eficiencia del almacenamiento.

### Restricciones

- Solo se admiten clústeres de v1.15 o posterior.
- El clúster debe utilizar el complemento everest de la versión 1.1.13 o posterior.
- No se admiten contenedores de Kata.
- Cuando se utiliza el complemento everest anterior a 1.2.69 o 2.1.11, se puede crear un máximo de 10 PVC simultáneamente mediante la función de subdirectorio. se recomienda everest de 1.2.69 o posterior o de 2.1.11 o posterior.

### Creación de un volumen de SFS Turbo del tipo de subpath

---

**⚠ ATENCIÓN**

No expanda, disocie ni elimine un volumen **subpath**.

---

**Paso 1** Cree un sistema de archivos de SFS Turbo en la misma VPC y subred que el clúster.

**Paso 2** Cree un archivo YAML de StorageClass como, por ejemplo, **sfsturbo-subpath-sc.yaml**.

A continuación, se presenta un ejemplo:

```
apiVersion: storage.k8s.io/v1
allowVolumeExpansion: true
kind: StorageClass
metadata:
  name: sfsturbo-subpath-sc
mountOptions:
- lock
parameters:
  csi.storage.k8s.io/csi-driver-name: sfsturbo.csi.everest.io
  csi.storage.k8s.io/fstype: nfs
  everest.io/archive-on-delete: "true"
  everest.io/share-access-to: 7ca2dba2-1234-1234-1234-626371a8fb3a
  everest.io/share-expand-type: bandwidth
  everest.io/share-export-location: 192.168.1.1:/sfsturbo/
  everest.io/share-source: sfs-turbo
  everest.io/share-volume-type: STANDARD
  everest.io/volume-as: subpath
  everest.io/volume-id: 0d773f2e-1234-1234-1234-de6a35074696
provisioner: everest-csi-provisioner
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

En este ejemplo:

- **name:** indica el nombre del StorageClass.
- **mountOptions:** indica las opciones de montaje. Este campo es opcional.
  - En versiones posteriores a everest 1.1.13 y anteriores a everest 1.2.8, solo se puede configurar el parámetro **nolock**. De forma predeterminada, se utiliza el parámetro **nolock** para la operación de montaje y no es necesario configurarlo. Si **nolock** se establece en **false**, se utiliza el campo **lock**.
  - A partir de everest 1.2.8, se admiten más opciones de montaje. Para obtener más información, consulte [Configuración de las opciones de montaje de SFS Turbo](#). **No ponga nolock a true. De lo contrario, la operación de montaje fallará.**

```
mountOptions:
- vers=3
- timeo=600
- nolock
- hard
```

- **everest.io/volume-as:** Este parámetro se establece en **subpath** para usar el volumen **subpath**.
- **everest.io/share-access-to:** Este parámetro es opcional. En un volumen **subpath**, establezca este parámetro en el ID de la VPC donde se encuentra el sistema de archivos de SFS Turbo.
- **everest.io/share-expand-type:** Este parámetro es opcional. Si el tipo del sistema de archivos de SFS Turbo es SFS Turbo Standard – Enhanced o SFS Turbo Performance – Enhanced, establezca este parámetro en **bandwidth**.
- **everest.io/share-export-location:** Este parámetro indica el directorio de montaje. Consiste en la ruta compartida de SFS Turbo y el subdirectorio. La ruta compartida se puede obtener en la consola de SFS Turbo. El subdirectorio está definido por el usuario. Los PVC creados con StorageClass se encuentran en este subdirectorio.
- **everest.io/share-volume-type:** Este parámetro es opcional. Especifica el tipo de sistema de archivos de SFS Turbo. El valor puede ser **STANDARD** o **PERFORMANCE**. Para

los tipos mejorados, este parámetro debe usarse junto con **everest.io/share-expand-type** (cuyo valor debe ser **bandwidth**).

- **everest.io/zone**: Este parámetro es opcional. Establezca la AZ donde se encuentra el sistema de archivos de SFS Turbo.
- **everest.io/volume-id**: Este parámetro indica el ID del volumen de SFS Turbo. Puede obtener el ID de volumen en la página de SFS Turbo.
- **everest.io/archive-on-delete**: Si este parámetro se establece en **true** y se selecciona **Delete** para **Reclaim Policy**, los documentos originales del PV se archivarán en el directorio llamado **archived- $\{PV\ name.timestamp\}$**  antes de eliminar el PVC. Si este parámetro se establece en **false**, se eliminará el subdirectorio de SFS Turbo del PV correspondiente. El valor predeterminado es **true**, que indica que los documentos originales del PV se archivarán en el directorio llamado **archived- $\{PV\ name.timestamp\}$**  antes de eliminar el PVC.

**Paso 3** Ejecute **kubectl create -f sfsturbo-subpath-sc.yaml**.

**Paso 4** Cree un archivo YAML de PVC denominado **sfs-turbo-test.yaml**.

A continuación, se presenta un ejemplo:

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: sfs-turbo-test
  namespace: default
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 50Gi
  storageClassName: sfsturbo-subpath-sc
  volumeMode: Filesystem
```

En este ejemplo:

- **name**: indica el nombre del PVC.
- **storageClassName**: especifica el nombre del StorageClass creado en el paso anterior.
- **storage**: En el modo de ruta secundaria, es inútil especificar este parámetro. La capacidad de almacenamiento está limitada por la capacidad total del sistema de archivos de SFS Turbo. Si la capacidad total del sistema de archivos de SFS Turbo es insuficiente, amplíe la capacidad en la página SFS Turbo de manera oportuna.

**Paso 5** Ejecute el comando **kubectl create -f sfs-turbo-test.yaml** para crear un PVC.

----Fin

#### NOTA

No tiene sentido llevar a cabo la expansión de la capacidad en un volumen de SFS Turbo creado en el modo de ruta secundaria. Esta operación no amplía la capacidad del sistema de archivos de SFS Turbo. Asegúrese de que la capacidad total del sistema de archivos de SFS Turbo no se agote.

## Creación de un Deployment y montaje de un volumen existente

**Paso 1** Cree un archivo YAML para el Deployment, por ejemplo, **deployment-test.yaml**.

A continuación, se presenta un ejemplo:

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: test-turbo-subpath-example
  namespace: default
  generation: 1
  labels:
    appgroup: ''
spec:
  replicas: 1
  selector:
    matchLabels:
      app: test-turbo-subpath-example
  template:
    metadata:
      labels:
        app: test-turbo-subpath-example
    spec:
      containers:
        - image: nginx:latest
          name: container-0
          volumeMounts:
            - mountPath: /tmp
              name: pvc-sfs-turbo-example
          restartPolicy: Always
          imagePullSecrets:
            - name: default-secret
          volumes:
            - name: pvc-sfs-turbo-example
              persistentVolumeClaim:
                claimName: sfs-turbo-test
    
```

En este ejemplo:

- **name**: indica el nombre del Deployment.
- **image**: especifica la imagen utilizada por el Deployment.
- **mountPath**: indica el camino de montaje del contenedor. En este ejemplo, el volumen se monta en el directorio **/tmp**.
- **claimName**: indica el nombre de un PVC existente.

**Paso 2** Cree el Deployment.

**kubectl create -f deployment-test.yaml**

----Fin

## Creación dinámica de un volumen de ruta secundaria para un StatefulSet

**Paso 1** Cree un archivo YAML para un StatefulSet como, por ejemplo, **StatefulSet-test.yaml**.

A continuación, se presenta un ejemplo:

```

apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: test-turbo-subpath
  namespace: default
  generation: 1
  labels:
    appgroup: ''
spec:
  replicas: 2
  selector:
    matchLabels:
      app: test-turbo-subpath
    
```

```
template:
  metadata:
    labels:
      app: test-turbo-subpath
    annotations:
      metrics.alpha.kubernetes.io/custom-endpoints:
      '[{"api":"","path":"","port":"","names":""}]'
      pod.alpha.kubernetes.io/initialized: 'true'
  spec:
    containers:
      - name: container-0
        image: 'nginx:latest'
        resources: {}
        volumeMounts:
          - name: sfs-turbo-160024548582479676
            mountPath: /tmp
            terminationMessagePath: /dev/termination-log
            terminationMessagePolicy: File
            imagePullPolicy: IfNotPresent
        restartPolicy: Always
        terminationGracePeriodSeconds: 30
        dnsPolicy: ClusterFirst
        securityContext: {}
        imagePullSecrets:
          - name: default-secret
        affinity: {}
        schedulerName: default-scheduler
    volumeClaimTemplates:
      - metadata:
          name: sfs-turbo-160024548582479676
          namespace: default
          annotations: {}
        spec:
          accessModes:
            - ReadWriteOnce
          resources:
            requests:
              storage: 10Gi
            storageClassName: sfsturbo-subpath-sc
          serviceName: www
          podManagementPolicy: OrderedReady
          updateStrategy:
            type: RollingUpdate
          revisionHistoryLimit: 10
```

En este ejemplo:

- **name:** indica el nombre del StatefulSet.
- **image:** especifica la imagen utilizada por el StatefulSet.
- **mountPath:** indica el camino de montaje del contenedor. En este ejemplo, el volumen se monta en el directorio **/tmp**.
- **spec.template.spec.contenedores.volumeMounts.name** y **spec.volumeClaimTemplates.metadata.name** deben ser coherentes porque tienen una relación de asignación.
- **storageClassName:** indica el nombre del StorageClass.

**Paso 2** Cree el StatefulSet.

```
kubectl create -f statefulset-test.yaml
```

----Fin

## 8.6 Object Storage Service (OBS)

### 8.6.1 Descripción general

#### Presentación

Object Storage Service (OBS) proporciona capacidades de almacenamiento de datos masivas, seguras y rentables para que almacene datos de cualquier tipo y tamaño. Puede usarlo en backup/archivo empresarial, video bajo demanda (VoD), videovigilancia y muchos otros escenarios.

- **Standard APIs:** Con las API RESTful de HTTP, OBS le permite usar herramientas de cliente o herramientas de terceros para acceder al almacenamiento de objetos.
- **Data sharing:** Los servidores, los dispositivos integrados y los dispositivos IoT pueden usar la misma ruta para acceder a los datos de objetos compartidos en OBS.
- **Public/Private networks:** OBS permite acceder a los datos desde las redes públicas para cumplir con los requisitos de las aplicaciones de Internet.
- **Capacity and performance:** Sin límite de capacidad; alto rendimiento (latencia de E/S de lectura/escritura dentro de 10 ms).
- **Use cases:** Deployments/StatefulSets en modo **ReadOnlyMany** y trabajos creados para análisis de big data, alojamiento de sitios web estático, VOD en línea, secuenciación de genes, videovigilancia inteligente, copia de respaldo y archivo, y cajas en la nube empresarial (discos web). Puede crear almacenamiento de objetos mediante la consola de OBS, las herramientas y los SDK.

#### Especificaciones de OBS

OBS proporciona múltiples clases de almacenamiento para satisfacer los requisitos de los clientes en cuanto al rendimiento y los costos de almacenamiento.

- **Parallel File System (PFS, **recomendado**):** Es un sistema de archivos optimizado de alto rendimiento proporcionado por OBS. Proporciona latencia de acceso de nivel de milisegundos, ancho de banda de nivel TB/s e IOPS de nivel de millones, y puede procesar rápidamente cargas de trabajo de HPC. PFS supera a los buckets de OBS. Para obtener más información, consulte [Acerca del Parallel File System](#).
- **Bucket de objetos (**no recomendado**):**
  - **Standard:** cuenta con baja latencia y alto rendimiento. Por lo tanto, es bueno para almacenar con frecuencia (varias veces al mes) archivos accedidos o archivos pequeños (menos de 1 MB). Los escenarios de aplicación incluyen análisis de big data, aplicaciones móviles, videos calientes y aplicaciones sociales.
  - **OBS Infrequent Access:** aplicable para almacenar datos de acceso semifrecuente (menos de 12 veces al año) que requieren una respuesta rápida. Sus escenarios de aplicación incluyen sincronización o uso compartido de archivos y copia de respaldo a nivel empresarial. Esta clase de almacenamiento tiene la misma durabilidad, baja latencia y alto rendimiento que la clase de almacenamiento estándar, con un costo menor, pero su disponibilidad es ligeramente menor que la clase de almacenamiento estándar.

Para obtener más información acerca de las clases de almacenamiento de OBS, consulte [Clases de almacenamiento](#).

## Escenario

OBS admite los siguientes modos de montaje basados en escenarios de aplicación:

- **Uso de un bucket de OBS existente con un PV estático:** modo de creación estática, donde se utiliza un volumen de OBS existente para crear un PV y luego montar el almacenamiento en la carga de trabajo con un PVC. Este modo es aplicable a escenarios en los que el almacenamiento subyacente está disponible o se factura anualmente/mensualmente.
- **Uso de un bucket de OBS con un PV dinámico:** modo de creación dinámica, donde no es necesario crear volúmenes de OBS por adelantado. En su lugar, especifique un StorageClass durante la creación de PVC y se creará automáticamente un volumen de OBS y un PV. Este modo es aplicable a escenarios en los que no hay almacenamiento subyacente disponible.

## Restricciones

- Los contenedores de Kata no admiten los volúmenes de OBS.
- Un solo usuario puede crear un máximo de 100 bucket de OBS en la consola. Si tiene una gran cantidad de cargas de trabajo de CCE y desea montar un bucket de OBS en cada carga de trabajo, puede que se quede sin bucket fácilmente. En este escenario, se recomienda usar OBS con la API o el SDK de OBS y no montar bucket de OBS en la carga de trabajo de la consola.
- Cuando se utilizan sistemas de archivos paralelos y bucket de objetos, el grupo y el permiso del punto de montaje no se pueden modificar.
- CCE le permite utilizar sistemas de archivos paralelos de OBS llamando al SDK de OBS o a través del montaje de PVC. El montaje de PVC es implementado por el **obsfs tool** proporcionado por OBS. Para obtener más información sobre obsfs, consulte la [Introducción a obsfs](#). Cada vez que se monta un sistema de archivos paralelos de OBS, se genera un proceso residente obsfs, como se muestra en la siguiente figura.

```
[root@cluster-1198-prr-38854 ~]# ps -aux | grep obsfs
root      753  0.0  0.1 93520 4688 ?        Ssl  11:09   0:00 /usr/bin/obsfs pvc-e17bf9a8-3367-4814-9a23-fbaa55693cf1 /mnt/passthrough/huawei.com:443 -o endpoint=cn-north-7 -o passwd_file=/opt/everest-host-connector/1517937763902500338_obsfsprod/pvc-e17bf9a8-3367-4814-9a23-fbaa55693cf1 -o allow_other -o nonempty -o big_writes -o max_write=131072 -o max_background=100 -o use_ino -o no_check_certificate -o umask=0
```

Se recomienda reservar 1 GB de memoria para cada proceso de obsfs. Por ejemplo, para un nodo con 4 CPUs y 8 GB de memoria, el sistema de archivos paralelo obsfs debe montarse en **no más de ocho pods**.

### NOTA

Los procesos residentes de obsfs se ejecutan en el nodo. Si la memoria consumida excede el límite superior del nodo, el nodo se vuelve anormal. En un nodo con 4 CPU y 8 GB de memoria, si se montan más de 100 pods en sistemas de archivos paralelos, el nodo dejará de estar disponible. Se recomienda controlar el número de pods montados en sistemas de archivos paralelos en un solo nodo.

## Facturación

- Cuando se monta un volumen de OBS, el modo de facturación de OBS **creado automáticamente** con StorageClass es de **pago por uso** de forma predeterminada. Para obtener más información sobre los precios de OBS, consulte [Detalles de precios de OBS](#).



1. Elija **Storage** en el panel de navegación y haga clic en la ficha **PersistentVolumeClaims (PVCs)**. Haga clic en **Create PVC** en la esquina superior derecha. En el cuadro de diálogo que se muestra, configure los parámetros de PVC.

Parámetro	Descripción
PVC Type	En esta sección, seleccione <b>OBS</b> .
PVC Name	Escriba el nombre de PVC, que debe ser único en el mismo espacio de nombres.
Creation Method	<ul style="list-style-type: none"> <li>– Si el almacenamiento subyacente está disponible, puede crear un volumen de almacenamiento o utilizar un volumen de almacenamiento existente para crear estáticamente un PVC en función de si se ha creado un PV.</li> <li>– Si no hay ningún almacenamiento subyacente disponible, puede seleccionar <b>Dynamically provision</b>. Para obtener más información, véase <a href="#">Uso de un bucket de OBS con un PV dinámico</a>.</li> </ul> <p>En este ejemplo, seleccione <b>Create new</b> para crear un PV y un PVC al mismo tiempo en la consola.</p>
PV <sup>a</sup>	<p>Seleccione un volumen de PV existente en el clúster. Es necesario crear un PV por adelantado. Para obtener más información, consulte "Creación de un volumen de almacenamiento (PV)" de <a href="#">Operaciones relacionadas</a>.</p> <p>No es necesario especificar este parámetro en este ejemplo.</p>
OBS <sup>b</sup>	<p>Haga clic en <b>Select OBS</b>. En la página mostrada, seleccione el bucket de OBS que cumpla con sus requisitos y haga clic en <b>OK</b>.</p> <p><b>NOTA</b>                      Actualmente, solo se admiten sistemas de archivos paralelos.</p>
PV Name <sup>b</sup>	Introduzca el nombre de PV, que debe ser único en el mismo clúster.
Access Mode <sup>b</sup>	Los volúmenes de OBS solo admiten <b>ReadWriteMany</b> , lo que indica que un volumen de almacenamiento puede montarse en múltiples nodos en modo de lectura/escritura. Para obtener más información, véase <a href="#">Modos de acceso al volumen</a> .
Reclaim Policy <sup>b</sup>	<p>Puede seleccionar <b>Delete</b> o <b>Retain</b> para especificar la política de recuperación del almacenamiento subyacente cuando se elimina el PVC. Para obtener más información, véase <a href="#">Política de reclamo de PV</a>.</p> <p><b>NOTA</b>                      Si varios PV utilizan el mismo volumen de OBS, utilice <b>Retain</b> para evitar la eliminación en cascada de volúmenes subyacentes.</p>

Parámetro	Descripción
Secret <sup>b</sup>	<p><b>Custom:</b> Personalice un secreto si desea asignar diferentes permisos de usuario a diferentes dispositivos de almacenamiento de OBS. Para obtener más información, véase <a href="#">Uso de una AK/SK personalizada para montar un volumen de OBS</a>.</p> <p>Solo se pueden seleccionar secretos con la etiqueta <b>secret.kubernetes.io/used-by = csi</b>. El tipo de secreto es cfe/secure-opaque. Si no hay ningún secreto disponible, haga clic en <b>Create Secret</b> para crear uno.</p> <ul style="list-style-type: none"> <li>– <b>Name:</b> Ingrese un nombre de secreto.</li> <li>– <b>Namespace:</b> Seleccione el espacio de nombres donde está el secreto.</li> <li>– <b>Access Key (AK/SK):</b> Sube un archivo clave en formato .csv. Para obtener más información, véase <a href="#">Obtención de una clave de acceso</a>.</li> </ul>
Mount Options <sup>b</sup>	<p>Introduzca los pares de clave-valor del parámetro de montaje. Para obtener más información, véase <a href="#">Configuración de opciones de montaje de OBS</a>.</p>

 **NOTA**

a: El parámetro está disponible cuando **Creation Method** se establece en **Use existing**.

b: El parámetro está disponible cuando **Creation Method** se establece en **Create new**.

2. Haga clic en **Create** para crear un PVC y un PV.

Puede elegir **Storage** en el panel de navegación y ver el PVC y PV creados en las páginas de fichas **PersistentVolumeClaims (PVCs)** y **PersistentVolumes (PVs)**.

**Paso 3** Cree una aplicación.

1. En el panel de navegación de la izquierda, haga clic en **Workloads**. En el panel derecho, haga clic en la ficha **Deployments**.
2. Haga clic en **Create Workload** en la esquina superior derecha. En la página mostrada, haga clic en **Data Storage** en el área **Container Settings** y haga clic en **Add Volume** para seleccionar **PVC**.

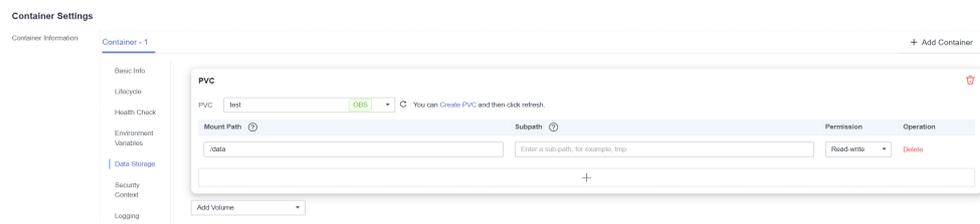
Montar y utilizar volúmenes de almacenamiento, como se muestra en [Tabla 8-29](#). Para obtener más información sobre otros parámetros, consulte [Workloads](#).

**Tabla 8-29** Montaje de un volumen de almacenamiento

Parámetro	Descripción
PVC	Seleccione un volumen de almacenamiento de objetos existente.

Parámetro	Descripción
Add Container Path	<p>Configure los parámetros siguientes:</p> <ol style="list-style-type: none"> <li><b>Mount Path:</b> Introduzca una ruta de acceso del contenedor. Por ejemplo, <code>/tmp</code>. Este parámetro indica la ruta de contenedor en la que se montará un volumen de datos. No monte el volumen en un directorio del sistema como <code>/</code> o <code>/var/run</code>; esta acción puede causar errores de contenedor. Se recomienda montar el volumen en un directorio vacío. Si el directorio no está vacío, asegúrese de que no hay archivos que afecten al inicio de contenedor. De lo contrario, se reemplazarán los archivos, lo que provocará errores de inicio de contenedor o errores de creación de carga de trabajo.                     <p><b>AVISO</b></p>                     Cuando el contenedor está montado en un directorio de alto riesgo, se recomienda utilizar una cuenta con los permisos mínimos para iniciar el contenedor; de lo contrario, los archivos de alto riesgo en la máquina host podrían estar dañados.</li> <li><b>Subpath:</b> Introduzca una ruta secundaria, por ejemplo, <code>tmp</code>. Se utiliza una ruta secundaria para montar un volumen local de modo que se utilice el mismo volumen de datos en un solo pod. Si este parámetro se deja en blanco, la ruta raíz se utiliza de forma predeterminada.</li> <li><b>Permission</b> <ul style="list-style-type: none"> <li><b>Read-only:</b> Solo puede leer los datos en los volúmenes montados.</li> <li><b>Read/Write:</b> Puede modificar los volúmenes de datos montados en la ruta de acceso. Los datos recién escritos no se migran si se migra el contenedor, lo que puede causar la pérdida de datos.</li> </ul> </li> </ol> <p>Puede hacer clic en <b>+</b> para agregar varias rutas y subrutas.</p>

En este ejemplo, el disco se monta en la trayectoria `/data` del contenedor. Los datos de contenedor generados en esta ruta se almacenan en el volumen de OBS.



3. Después de completar la configuración, haga clic en **Create**.

Una vez creada la carga de trabajo, puede probar [Verificación de la persistencia y el uso compartido de datos](#).

----Fin

## (kubectl) Uso de un bucket de OBS existente

**Paso 1** Utilice kubectl para conectarse al clúster.

**Paso 2** Cree un PV.

1. Cree el archivo **pv-obs.yaml**.

```

apiVersion: v1
kind: PersistentVolume
metadata:
  annotations:
    pv.kubernetes.io/provisioned-by: everest-csi-provisioner
    everest.io/reclaim-policy: retain-volume-only # (Optional) The PV is
    deleted while the underlying volume is retained.
  name: pv-obs # PV name.
spec:
  accessModes:
    - ReadWriteMany # Access mode. The value must be ReadWriteMany for OBS.
  capacity:
    storage: 1Gi # OBS volume capacity.
  csi:
    driver: obs.csi.everest.io # Dependent storage driver for the
    mounting.
    driver: obs.csi.everest.io # Instance type.
    volumeHandle: <your_volume_id> # Name of the OBS volume.
    volumeAttributes:
      storage.kubernetes.io/csiProvisionerIdentity: everest-csi-provisioner
      everest.io/obs-volume-type: STANDARD
      everest.io/region: <your_region> # Region where
      the OBS volume is.
      everest.io/enterprise-project-id: <your_project_id> # (Optional)
      Enterprise project ID. If an enterprise project is specified, you need to use
      the same enterprise project when creating a PVC. Otherwise, the PVC cannot be
      bound to a PV.
    nodePublishSecretRef: # Custom secret of the OBS volume.
      name: <your_secret_name> # Custom secret name.
      namespace: <your_namespace> # Namespace of the custom secret.
    persistentVolumeReclaimPolicy: Retain # Reclaim policy.
    storageClassName: csi-obs # Storage class name.
    mountOptions: [] # Mount options.
    
```

**Tabla 8-30** Parámetros clave

Parámetro	Obligatorio	Descripción
everest.io/reclaim-policy: retain-volume-only	No	Opcional. Actualmente, solo se admite <b>retain-volume-only</b> . Este campo solo es válido cuando la versión más reciente es 1.2.9 o posterior y la política de recuperación es <b>Delete</b> . Si la política de reclamación es de <b>Delete</b> y el valor actual es de <b>retain-volume-only</b> el PV asociado se elimina mientras se conserva el volumen de almacenamiento subyacente cuando se elimina un PVC.

Parámetro	Obligatorio	Descripción
fsType	Sí	Tipo de instancia. El valor puede ser <b>obsfs</b> o <b>s3fs</b> . <ul style="list-style-type: none"> <li>– <b>obsfs</b>: Sistema de archivos paralelo, que se monta usando obsfs (recomendado).</li> <li>– <b>s3fs</b>: Bucket de objetos, que se monta usando s3fs.</li> </ul>
volumeHandle	Sí	Nombre del volumen de OBS.
everest.io/obs-volume-type	Sí	Clase de almacenamiento de OBS. <ul style="list-style-type: none"> <li>– Si <b>fsType</b> se establece en <b>s3fs</b>, <b>STANDARD</b> (bucket estándar) y <b>WARM</b> (bucket de acceso poco frecuente) son compatibles.</li> <li>– Este parámetro no es válido cuando <b>fsType</b> se establece en <b>obsfs</b>.</li> </ul>
everest.io/region	Sí	Región donde se despliega el bucket de OBS. Para obtener más información sobre el valor de <b>region</b> , consulte <a href="#">Regiones y puntos de conexión</a> .
everest.io/enterprise-project-id	No	Opcional. ID de proyecto de empresa de OBS. Si se especifica un proyecto de empresa, debe especificar el mismo proyecto de empresa al crear un PVC. De lo contrario, el PVC no puede estar unido a un PV.  <b>Cómo obtenerlo:</b> En la consola de OBS, seleccione <b>Buckets</b> o <b>Parallel File Systems</b> en el panel de navegación de la izquierda. Haga clic en el nombre del bucket de OBS para acceder a su página de detalles. En el área <b>Basic Information</b> , busque el proyecto de empresa y haga clic en él para acceder a la consola del proyecto de empresa. Copie el ID correspondiente para obtener el ID del proyecto de empresa al que pertenece el almacenamiento de objetos.
nodePublishSecretRef	No	Clave de acceso (AK/SK) utilizada para montar el volumen de almacenamiento de objetos. Puede utilizar la AK/SK para crear un secreto y montarlo en el PV. Para obtener más información, véase <a href="#">Uso de una AK/SK personalizada para montar un volumen de OBS</a> .  Un ejemplo es el siguiente: <pre>nodePublishSecretRef:   name: secret-demo   namespace: default</pre>

Parámetro	Obligatorio	Descripción
mountOptions	No	Opciones de montaje. Para obtener más información, véase <a href="#">Configuración de opciones de montaje de OBS</a> .
persistentVolumeReclaimPolicy	Sí	<p>Se admite una política de recuperación cuando la versión del clúster es o posterior a 1.19.10 y la versión everest es o posterior a 1.2.9.</p> <p>Las políticas de recuperación <b>Delete</b> y <b>Retain</b> son compatibles. Para obtener más información, véase <a href="#">Política de reclamo de PV</a>. Si varios PV utilizan el mismo volumen de OBS, utilice <b>Retain</b> para evitar la eliminación en cascada de volúmenes subyacentes.</p> <p><b>Delete:</b></p> <ul style="list-style-type: none"> <li>– Si no se especifica <b>everest.io/reclaim-policy</b>, tanto el PV como los recursos de almacenamiento se eliminan cuando se elimina un PVC.</li> <li>– Si <b>everest.io/reclaim-policy</b> se establece en <b>retain-volume-only</b> cuando se elimina un PVC, se elimina el PV pero se conservan los recursos de almacenamiento.</li> </ul> <p><b>Retain:</b> Cuando se elimina un PVC, el PV y los recursos de almacenamiento subyacentes no se eliminan. En su lugar, debe eliminar manualmente estos recursos. Después de eso, el PV está en el estado <b>Released</b> y no puede estar ligado al PVC de nuevo.</p>
storage	Sí	<p>Capacidad de almacenamiento, en Gi.</p> <p>Para los bucket de OBS, este campo se utiliza solo para la verificación (no puede estar vacío o 0). Su valor se fija en <b>1</b> y cualquier valor que establezca no tiene efecto para los bucket de OBS.</p>
storageClassName	Sí	El nombre de clase de almacenamiento de los volúmenes de OBS es <b>csi-obs</b> .

2. Ejecute el siguiente comando para crear un PV:

```
kubectl apply -f pv-obs.yaml
```

### Paso 3 Cree un PVC.

1. Cree el archivo **pvc-obs.yaml**.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-obs
  namespace: default
  annotations:
```

```

volume.beta.kubernetes.io/storage-provisioner: everest-csi-provisioner
everest.io/obs-volume-type: STANDARD
csi.storage.k8s.io/fstype: obsfs
csi.storage.k8s.io/node-publish-secret-name: <your_secret_name> # Custom
secret name.
csi.storage.k8s.io/node-publish-secret-namespace: <your_namespace>
# Namespace of the custom secret.
everest.io/enterprise-project-id: <your_project_id> # (Optional)
Enterprise project ID. If an enterprise project is specified, you need to use
the same enterprise project when creating a PVC. Otherwise, the PVC cannot be
bound to a PV.
spec:
  accessModes:
    - ReadWriteMany # The value must be ReadWriteMany for OBS.
  resources:
    requests:
      storage: 1Gi
  storageClassName: csi-obs # Storage class name, which must be the
same as that of the PV.
  volumeName: pv-obs # PV name.

```

Tabla 8-31 Parámetros clave

Parámetro	Obligatorio	Descripción
csi.storage.k8s.io/node-publish-secret-name	No	Nombre del secreto personalizado especificado en el PV.
csi.storage.k8s.io/node-publish-secret-namespace	No	Espacio de nombres del secreto personalizado especificado en el PV.
everest.io/enterprise-project-id	No	ID del proyecto de OBS. <b>Cómo obtenerlo:</b> En la consola de OBS, seleccione <b>Buckets</b> o <b>Parallel File Systems</b> en el panel de navegación de la izquierda. Haga clic en el nombre del bucket de OBS para acceder a su página de detalles. En el área <b>Basic Information</b> , busque el proyecto de empresa y haga clic en él para acceder a la consola del proyecto de empresa. Copie el ID correspondiente para obtener el ID del proyecto de empresa al que pertenece el almacenamiento de objetos.
storage	Sí	Capacidad solicitada en el PVC, en Gi. Para OBS, este campo se utiliza solo para verificación (no puede estar vacío o 0). Su valor se fija en <b>1</b> y cualquier valor que establezca no tiene efecto para OBS.
storageClassName	Sí	Nombre de la clase de almacenamiento, que debe ser el mismo que la clase de almacenamiento del PV en <b>1</b> . El nombre de clase de almacenamiento de los volúmenes de OBS es <b>csi-obs</b> .

Parámetro	Obligatorio	Descripción
volumeName	Sí	Nombre de PV, que debe ser el mismo que el nombre de PV en <b>1</b> .

- Ejecute el siguiente comando para crear un PVC:

```
kubectl apply -f pvc-obs.yaml
```

#### Paso 4 Cree una aplicación.

1. Cree un archivo denominado **web-demo.yaml**. En este ejemplo, el volumen de OBS se monta en la ruta **/data**.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web-demo
  namespace: default
spec:
  replicas: 2
  selector:
    matchLabels:
      app: web-demo
  template:
    metadata:
      labels:
        app: web-demo
    spec:
      containers:
      - name: container-1
        image: nginx:latest
        volumeMounts:
        - name: pvc-obs-volume #Volume name, which must be the same as the
          volume name in the volumes field.
          mountPath: /data #Location where the storage volume is mounted.
        imagePullSecrets:
        - name: default-secret
      volumes:
      - name: pvc-obs-volume #Volume name, which can be customized.
        persistentVolumeClaim:
          claimName: pvc-obs #Name of the created PVC.
```

2. Ejecute el siguiente comando para crear una aplicación en la que esté montado el volumen de OBS:

```
kubectl apply -f web-demo.yaml
```

Una vez creada la carga de trabajo, puede probar [Verificación de la persistencia y el uso compartido de datos](#).

----Fin

## Verificación de la persistencia y el uso compartido de datos

### Paso 1 Vea las aplicaciones y los archivos desplegados.

1. Ejecute el siguiente comando para ver los pods creados:

```
kubectl get pod | grep web-demo
```

Producto esperado:

```
web-demo-846b489584-mjhm9 1/1 Running 0 46s
web-demo-846b489584-wvv5s 1/1 Running 0 46s
```

2. Ejecute los siguientes comandos en secuencia para ver los archivos en la ruta **/data** de los pods:

```
kubectl exec web-demo-846b489584-mjhm9 -- ls /data
kubectl exec web-demo-846b489584-wvv5s -- ls /data
```

Si no se devuelve ningún resultado para ambos pods, no existe ningún archivo en la ruta **/data**.

**Paso 2** Ejecute el siguiente comando para crear un archivo llamado **static** en la ruta **/data**:

```
kubectl exec web-demo-846b489584-mjhm9 -- touch /data/static
```

**Paso 3** Ejecute el siguiente comando para ver los archivos en la ruta **/data**:

```
kubectl exec web-demo-846b489584-mjhm9 -- ls /data
```

Producto esperado:

```
static
```

**Paso 4 Verificar la persistencia de los datos.**

1. Ejecute el siguiente comando para eliminar el pod llamado **web-demo-846b489584-mjhm9**:

```
kubectl delete pod web-demo-846b489584-mjhm9
```

Producto esperado:

```
pod "web-demo-846b489584-mjhm9" deleted
```

Después de la eliminación, el controlador de Deployment crea automáticamente una réplica.

2. Ejecute el siguiente comando para ver el pod creado:

```
kubectl get pod | grep web-demo
```

El resultado esperado es el siguiente, en el que **web-demo-846b489584-d4d4j** es el pod recién creado:

```
web-demo-846b489584-d4d4j 1/1 Running 0 110s
web-demo-846b489584-wvv5s 1/1 Running 0 7m50s
```

3. Ejecute el siguiente comando para comprobar si se han modificado los archivos de la ruta **/data** del nuevo pod:

```
kubectl exec web-demo-846b489584-d4d4j -- ls /data
```

Producto esperado:

```
static
```

Si el archivo **static** todavía existe, los datos se pueden almacenar de forma persistente.

**Paso 5 Verifique el uso compartido de datos.**

1. Ejecute el siguiente comando para ver el pod creado:

```
kubectl get pod | grep web-demo
```

Producto esperado:

```
web-demo-846b489584-d4d4j 1/1 Running 0 7m
web-demo-846b489584-wvv5s 1/1 Running 0 13m
```

2. Ejecute el siguiente comando para crear un archivo llamado **share** en la ruta **/data** de cualquier pod: En este ejemplo, seleccione el pod llamado **web-demo-846b489584-d4d4j**.

```
kubectl exec web-demo-846b489584-d4d4j -- touch /data/share
```

Compruebe los archivos en la ruta **/data** del pod.

```
kubectl exec web-demo-846b489584-d4d4j -- ls /data
```

Producto esperado:

```
share
static
```

3. Compruebe si el archivo **share** existe en la ruta **/data** de otro pod (**web-demo-846b489584-wvv5s**) también para verificar el uso compartido de datos.

```
kubectl exec web-demo-846b489584-wvv5s -- ls /data
```

Producto esperado:

```
share
static
```

Después de crear un archivo en la ruta **/data** de un pod, si el archivo también se crea en la ruta **/data** de otros pods, los dos pods comparten el mismo volumen.

----Fin

## Operaciones relacionadas

También puede realizar las operaciones que aparecen en [Tabla 8-32](#).

**Tabla 8-32** Operaciones relacionadas

Operación	Descripción	Procedimiento
Creación de un volumen de almacenamiento (PV)	Cree un PV en la consola de CCE.	<ol style="list-style-type: none"> <li data-bbox="833 371 1426 1803"> <p>Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumes (PVs)</b>. Haga clic en <b>Create Volume</b> en la esquina superior derecha. En el cuadro de diálogo que se muestra, configure los parámetros.</p> <ul style="list-style-type: none"> <li data-bbox="874 551 1273 584">● <b>Volume Type:</b> Seleccione <b>OBS</b>.</li> <li data-bbox="874 595 1410 723">● <b>OBS:</b> Haga clic en <b>Select OBS</b>. En la página mostrada, seleccione el almacenamiento de OBS que cumpla con sus requisitos y haga clic en <b>OK</b>.</li> <li data-bbox="874 741 1410 804">● <b>PV Name:</b> Introduzca el nombre de PV, que debe ser único en el mismo clúster.</li> <li data-bbox="874 819 1422 1014">● <b>Access Mode:</b> Los volúmenes de SFS solo admiten <b>ReadWriteMany</b>, lo que indica que un volumen de almacenamiento se puede montar en varios nodos en modo de lectura/escritura. Para obtener más información, véase <a href="#">Modos de acceso al volumen</a>.</li> <li data-bbox="874 1032 1394 1126">● <b>Reclaim Policy:</b> <b>Delete</b> o <b>Retain</b>. Para obtener más información, véase <a href="#">Modos de acceso al volumen</a>.</li> </ul> <p><b>NOTA</b>                      Si varios PV utilizan el mismo volumen de almacenamiento subyacente, utilice <b>Retain</b> para evitar la eliminación en cascada de volúmenes subyacentes.</p> <ul style="list-style-type: none"> <li data-bbox="874 1296 1426 1491">● <b>Custom:</b> Personalice un secreto si desea asignar diferentes permisos de usuario a diferentes dispositivos de almacenamiento de OBS. Para obtener más información, véase <a href="#">Uso de una AK/SK personalizada para montar un volumen de OBS</a>. Solo se pueden seleccionar secretos con la etiqueta <b>secret.kubernetes.io/used-by = csi</b>. El tipo de secreto es <b>cfe/secure-opaque</b>. Si no hay ningún secreto disponible, haga clic en <b>Create Secret</b> para crear uno.</li> <li data-bbox="874 1675 1426 1803">● <b>Mount Options:</b> Introduzca los pares clave-valor del parámetro de montaje. Para obtener más información, véase <a href="#">Configuración de opciones de montaje de OBS</a>.</li> </ul> </li> <li data-bbox="833 1821 1107 1854">2. Haga clic en <b>Create</b>.</li> </ol>

Operación	Descripción	Procedimiento
Actualización de una clave de acceso	Actualice la clave de acceso del almacenamiento de objetos en la consola de CCE.	<ol style="list-style-type: none"> <li>1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumeClaims (PVCs)</b>. Haga clic en <b>More &gt; Update Access Key</b> en la columna <b>Operation</b> del PVC.</li> <li>2. Sube un archivo clave en formato .csv. Para obtener más información, véase <a href="#">Obtención de una clave de acceso</a>. Haga clic en <b>OK</b>.</li> </ol> <p><b>NOTA</b>                      Después de actualizar una clave de acceso global, se puede acceder a todos los pods montados con el almacenamiento de objetos que utiliza esta clave de acceso solo después de reiniciarse.</p>
Eventos	Puede ver los nombres de eventos, los tipos de eventos, el número de ocurrencias, los eventos de Kubernetes, la hora de primera ocurrencia y la hora de última ocurrencia del PVC o PV.	<ol style="list-style-type: none"> <li>1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumeClaims (PVCs)</b> o <b>PersistentVolumes (PVs)</b>.</li> <li>2. Haga clic en <b>View Events</b> en la columna <b>Operation</b> del PVC o del PV de destino para ver los eventos generados en una hora (los datos de eventos se conservan durante una hora).</li> </ol>
Consulta de un archivo YAML	Puede ver, copiar y descargar los archivos YAML de un PVC o PV.	<ol style="list-style-type: none"> <li>1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumeClaims (PVCs)</b> o <b>PersistentVolumes (PVs)</b>.</li> <li>2. Haga clic en <b>View YAML</b> en la columna <b>Operation</b> del PVC o del PV de destino para ver o descargar el YAML.</li> </ol>

### 8.6.3 Uso de un bucket de OBS con un PV dinámico

Esta sección describe cómo crear automáticamente un bucket de OBS. Es aplicable cuando no hay ningún volumen de almacenamiento subyacente disponible.

#### Restricciones

- Los contenedores de Kata no admiten los volúmenes de OBS.
- Un solo usuario puede crear un máximo de 100 bucket de OBS en la consola. Si tiene una gran cantidad de cargas de trabajo de CCE y desea montar un bucket de OBS en cada carga de trabajo, puede que se quede sin bucket fácilmente. En este escenario, se recomienda usar OBS con la API o el SDK de OBS y no montar bucket de OBS en la carga de trabajo de la consola.
- Cuando se utilizan sistemas de archivos paralelos y bucket de objetos, el grupo y el permiso del punto de montaje no se pueden modificar.
- CCE le permite utilizar sistemas de archivos paralelos de OBS llamando al SDK de OBS o a través del montaje de PVC. El montaje de PVC es implementado por el **obsfs tool** proporcionado por OBS. Para obtener más información sobre obsfs, consulte la

**Introducción a obsfs.** Cada vez que se monta un sistema de archivos paralelos de OBS, se genera un proceso residente obsfs, como se muestra en la siguiente figura.

```

[root@k8s-master-1198-prf-50064 ~]# ps -aux | grep obsfs
root      7593  0.0  0.1 532580 44848  ?        Ssl  11:09   0:00 /usr/bin/obsfs pvc-e176f8a8-3367-4814-9a23-fbaa55093cf1 /mnt/paas/kubernetes/kubelet/pods/46899582-36ac-41b0-b684-bb32e168d01/volumes/kubernetes.io~csi/pvc-e176f8a8-3367-4814-9a23-fbaa55093cf1/mount --url=https://obs.cn-north-7.ultraqab.huaweicloud.com:443 --endpoint=cn-north-7 --passwd=/etc/everest-host-connector/161793763992500336_obsmpcred/pvc-e176f8a8-3367-4814-9a23-fbaa55093cf1 --o
1198-obsfs --noexec --no write --max_writes=1072 --max_backround=100 --log_dir --no_check_certificate --passkey
    
```

Se recomienda reservar 1 GB de memoria para cada proceso de obsfs. Por ejemplo, para un nodo con 4 CPUs y 8 GB de memoria, el sistema de archivos paralelo obsfs debe montarse en **no más de ocho pods**.

**NOTA**

Los procesos residentes de obsfs se ejecutan en el nodo. Si la memoria consumida excede el límite superior del nodo, el nodo se vuelve anormal. En un nodo con 4 CPU y 8 GB de memoria, si se montan más de 100 pods en sistemas de archivos paralelos, el nodo dejará de estar disponible. Se recomienda controlar el número de pods montados en sistemas de archivos paralelos en un solo nodo.

## Creación automática de un volumen de OBS en la consola

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** Cree dinámicamente un PVC y un PV.

1. Elija **Storage** en el panel de navegación y haga clic en la ficha **PersistentVolumeClaims (PVCs)**. Haga clic en **Create PVC** en la esquina superior derecha. En el cuadro de diálogo que se muestra, configure los parámetros de PVC.

Parámetro	Descripción
PVC Type	En esta sección, seleccione <b>OBS</b> .
PVC Name	Escriba el nombre de PVC, que debe ser único en el mismo espacio de nombres.
Creation Method	<ul style="list-style-type: none"> <li>– Si no hay almacenamiento subyacente disponible, puede seleccionar <b>Dynamically provision</b> para crear un almacenamiento de PVC, PV y subyacente en la consola en modo en cascada.</li> <li>– Si el almacenamiento subyacente está disponible, puede crear un volumen de almacenamiento o utilizar un volumen de almacenamiento existente para crear estáticamente un PVC en función de si se ha creado un PV. Para obtener más información, véase <a href="#">Uso de un bucket de OBS existente con un PV estático</a>.</li> </ul> <p>En este ejemplo, seleccione <b>Dynamically provision</b>.</p>
Storage Classes	La clase de almacenamiento de los volúmenes de OBS es <b>csi-obs</b> .
Instance Type	<ul style="list-style-type: none"> <li>– <b>Parallel file system</b>: un sistema de archivos de alto rendimiento proporcionado por OBS. Proporciona latencia de acceso de nivel de milisegundos, ancho de banda de nivel TB/s y IOPS de nivel de millón. <b>Se recomiendan sistemas de archivos paralelos.</b></li> <li>– <b>Object bucket</b>: un contenedor que almacena objetos en OBS. Todos los objetos de un bucket están en el mismo nivel lógico.</li> </ul>

Parámetro	Descripción
OBS Class	Puede seleccionar los siguientes tipos de bucket de objetos: <ul style="list-style-type: none"> <li>– <b>Standard</b>: Aplicable cuando una gran cantidad de archivos de punto de acceso o archivos de tamaño pequeño necesitan ser accedidos con frecuencia (múltiples veces al mes en promedio) y requieren una respuesta de acceso rápido.</li> <li>– <b>Infrequent access</b>: Aplicable cuando no se accede con frecuencia a los datos (menos de 12 veces al año en promedio) pero requiere una respuesta de acceso rápido.</li> </ul>
Access Mode	Los volúmenes de OBS solo admiten <b>ReadWriteMany</b> , lo que indica que un volumen de almacenamiento puede montarse en múltiples nodos en modo de lectura/escritura. Para obtener más información, véase <a href="#">Modos de acceso al volumen</a> .
Secret	<p><b>Custom</b>: Personalice un secreto si desea asignar diferentes permisos de usuario a diferentes dispositivos de almacenamiento de OBS. Para obtener más información, véase <a href="#">Uso de una AK/SK personalizada para montar un volumen de OBS</a>.</p> <p>Solo se pueden seleccionar secretos con la etiqueta <b>secret.kubernetes.io/used-by = csi</b>. El tipo de secreto es <code>cfe/secure-opaque</code>. Si no hay ningún secreto disponible, haga clic en <b>Create Secret</b> para crear uno.</p> <ul style="list-style-type: none"> <li>– <b>Name</b>: Ingrese un nombre de secreto.</li> <li>– <b>Namespace</b>: Seleccione el espacio de nombres donde está el secreto.</li> <li>– <b>Access Key (AK/SK)</b>: Sube un archivo clave en formato <code>.csv</code>. Para obtener más información, véase <a href="#">Obtención de una clave de acceso</a>.</li> </ul>
Enterprise Project	Proyectos de empresa admitidos: predeterminado, al que pertenece el clúster o al que especifica la clase de almacenamiento.

2. Haga clic en **Create** para crear un PVC y un PV.

Puede elegir **Storage** en el panel de navegación y ver el PVC y PV creados en las páginas de fichas **PersistentVolumeClaims (PVCs)** y **PersistentVolumes (PVs)**.

**Paso 3** Cree una aplicación.

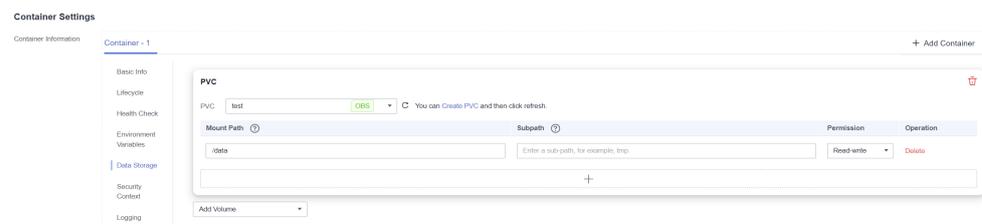
1. En el panel de navegación de la izquierda, haga clic en **Workloads**. En el panel derecho, haga clic en la ficha **Deployments**.
2. Haga clic en **Create Workload** en la esquina superior derecha. En la página mostrada, haga clic en **Data Storage** en el área **Container Settings** y haga clic en **Add Volume** para seleccionar **PVC**.

Montar y utilizar volúmenes de almacenamiento, como se muestra en [Tabla 8-33](#). Para obtener más información sobre otros parámetros, consulte [Workloads](#).

**Tabla 8-33** Montaje de un volumen de almacenamiento

Parámetro	Descripción
PVC	Seleccione un volumen de almacenamiento de objetos existente.
Add Container Path	<p>Configure los parámetros siguientes:</p> <ol style="list-style-type: none"> <li><b>Mount Path:</b> Introduzca una ruta de acceso del contenedor. Por ejemplo, <b>/tmp</b>.                      Este parámetro indica la ruta de contenedor en la que se montará un volumen de datos. No monte el volumen en un directorio del sistema como <b>/</b> o <b>/var/run</b>; esta acción puede causar errores de contenedor. Se recomienda montar el volumen en un directorio vacío. Si el directorio no está vacío, asegúrese de que no hay archivos que afecten al inicio de contenedor. De lo contrario, se reemplazarán los archivos, lo que provocará errores de inicio de contenedor o errores de creación de carga de trabajo.  <b>AVISO</b>                      Cuando el contenedor está montado en un directorio de alto riesgo, se recomienda utilizar una cuenta con los permisos mínimos para iniciar el contenedor; de lo contrario, los archivos de alto riesgo en la máquina host podrían estar dañados.</li> <li><b>Subpath:</b> Introduzca una ruta secundaria, por ejemplo, <b>tmp</b>. Se utiliza una ruta secundaria para montar un volumen local de modo que se utilice el mismo volumen de datos en un solo pod. Si este parámetro se deja en blanco, la ruta raíz se utiliza de forma predeterminada.</li> <li><b>Permission</b> <ul style="list-style-type: none"> <li>■ <b>Read-only:</b> Solo puede leer los datos en los volúmenes montados.</li> <li>■ <b>Read/Write:</b> Puede modificar los volúmenes de datos montados en la ruta de acceso. Los datos recién escritos no se migran si se migra el contenedor, lo que puede causar la pérdida de datos.</li> </ul> </li> </ol> <p>Puede hacer clic en <b>+</b> para agregar varias rutas y subrutas.</p>

En este ejemplo, el disco se monta en la trayectoria **/data** del contenedor. Los datos de contenedor generados en esta ruta se almacenan en el volumen de OBS.



3. Después de completar la configuración, haga clic en **Create**.

Una vez creada la carga de trabajo, puede probar [Verificación de la persistencia y el uso compartido de datos](#).

----Fin

## (kubectl) Creación automática de un volumen de OBS

**Paso 1** Utilice kubectl para conectarse al clúster.

**Paso 2** Utilice **StorageClass** para crear dinámicamente un PVC y un PV.

1. Cree el archivo **pvc-obs-auto.yaml**.

```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-obs-auto
  namespace: default
  annotations:
    everest.io/obs-volume-type: STANDARD # Object storage type.
    csi.storage.k8s.io/fstype: obsfs # Instance type.
    csi.storage.k8s.io/node-publish-secret-name: <your_secret_name> #
    Custom secret name.
    csi.storage.k8s.io/node-publish-secret-namespace: <your_namespace> #
    Namespace of the custom secret.
    everest.io/enterprise-project-id: <your_project_id> # (Optional)
    Enterprise project ID. If an enterprise project is specified, you need to use
    the same enterprise project when creating a PVC. Otherwise, the PVC cannot be
    bound to a PV.
spec:
  accessModes:
    - ReadWriteMany # For object storage, the value must be
    ReadWriteMany.
  resources:
    requests:
      storage: 1Gi # OBS volume capacity.
      storageClassName: csi-obs # The storage class type is OBS.
    
```

**Tabla 8-34** Parámetros clave

Parámetro	Obligatorio	Descripción
everest.io/obs-volume-type	Sí	Clase de almacenamiento de OBS. – Si <b>fsType</b> se establece en <b>s3fs</b> , <b>STANDARD</b> (bucket estándar) y <b>WARM</b> (bucket de acceso poco frecuente) son compatibles. – Este parámetro no es válido cuando <b>fsType</b> se establece en <b>obsfs</b> .
csi.storage.k8s.io/fstype	Sí	Tipo de instancia. El valor puede ser <b>obsfs</b> o <b>s3fs</b> . – <b>obsfs</b> : Sistema de archivos paralelo, que se monta usando obsfs (recomendado). – <b>s3fs</b> : Bucket de objetos, que se monta usando s3fs.

Parámetro	Obligatorio	Descripción
csi.storage.k8s.io/node-publish-secret-name	No	Nombre de secreto personalizado. (Recomendado) Seleccione esta opción si desea asignar diferentes permisos de usuario a diferentes dispositivos de almacenamiento de OBS. Para obtener más información, véase <a href="#">Uso de una AK/SK personalizada para montar un volumen de OBS</a> .
csi.storage.k8s.io/node-publish-secret-namespace	No	Espacio de nombres de un secreto personalizado.
everest.io/enterprise-project-id	No	ID del proyecto de OBS. <b>Cómo obtenerlo:</b> En la consola de OBS, seleccione <b>Buckets</b> o <b>Parallel File Systems</b> en el panel de navegación de la izquierda. Haga clic en el nombre del bucket de OBS para acceder a su página de detalles. En el área <b>Basic Information</b> , busque el proyecto de empresa y haga clic en él para acceder a la consola del proyecto de empresa. Copie el ID correspondiente para obtener el ID del proyecto de empresa al que pertenece el almacenamiento de objetos.
storage	Sí	Capacidad solicitada en el PVC, en Gi. Para los bucket de OBS, este campo se utiliza solo para la verificación (no puede estar vacío o 0). Su valor se fija en <b>1</b> y cualquier valor que establezca no tiene efecto para los bucket de OBS.
storageClassName	Sí	Nombre de clase de almacenamiento. El nombre de clase de almacenamiento de los volúmenes de OBS es <b>csi-obs</b> .

2. Ejecute el siguiente comando para crear un PVC:

```
kubectl apply -f pvc-obs-auto.yaml
```

### Paso 3 Cree una aplicación.

1. Cree un archivo denominado **web-demo.yaml**. En este ejemplo, el volumen de OBS se monta en la ruta **/data**.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web-demo
  namespace: default
spec:
  replicas: 2
  selector:
    matchLabels:
      app: web-demo
  template:
```

```

metadata:
  labels:
    app: web-demo
spec:
  containers:
  - name: container-1
    image: nginx:latest
    volumeMounts:
    - name: pvc-obs-volume      #Volume name, which must be the same as the
volume name in the volumes field.
      mountPath: /data      #Location where the storage volume is mounted.
    imagePullSecrets:
    - name: default-secret
  volumes:
  - name: pvc-obs-volume      #Volume name, which can be customized.
    persistentVolumeClaim:
      claimName: pvc-obs-auto      #Name of the created PVC.
    
```

2. Ejecute el siguiente comando para crear una aplicación en la que esté montado el volumen de OBS:

```
kubectl apply -f web-demo.yaml
```

Una vez creada la carga de trabajo, puede probar [Verificación de la persistencia y el uso compartido de datos](#).

----Fin

## Verificación de la persistencia y el uso compartido de datos

**Paso 1** Vea las aplicaciones y los archivos desplegados.

1. Ejecute el siguiente comando para ver los pods creados:

```
kubectl get pod | grep web-demo
```

Producto esperado:

```

web-demo-846b489584-mjhm9    1/1      Running    0          46s
web-demo-846b489584-wvv5s    1/1      Running    0          46s
    
```

2. Ejecute los siguientes comandos en secuencia para ver los archivos en la ruta **/data** de los pods:

```

kubectl exec web-demo-846b489584-mjhm9 -- ls /data
kubectl exec web-demo-846b489584-wvv5s -- ls /data
    
```

Si no se devuelve ningún resultado para ambos pods, no existe ningún archivo en la ruta **/data**.

**Paso 2** Ejecute el siguiente comando para crear un archivo llamado **static** en la ruta **/data**:

```
kubectl exec web-demo-846b489584-mjhm9 -- touch /data/static
```

**Paso 3** Ejecute el siguiente comando para ver los archivos en la ruta **/data**:

```
kubectl exec web-demo-846b489584-mjhm9 -- ls /data
```

Producto esperado:

```
static
```

**Paso 4** Verificar la persistencia de los datos.

1. Ejecute el siguiente comando para eliminar el pod llamado **web-demo-846b489584-mjhm9**:

```
kubectl delete pod web-demo-846b489584-mjhm9
```

Producto esperado:

```
pod "web-demo-846b489584-mjhm9" deleted
```

Después de la eliminación, el controlador de Deployment crea automáticamente una réplica.

2. Ejecute el siguiente comando para ver el pod creado:

```
kubectl get pod | grep web-demo
```

El resultado esperado es el siguiente, en el que **web-demo-846b489584-d4d4j** es el pod recién creado:

```
web-demo-846b489584-d4d4j 1/1 Running 0 110s
web-demo-846b489584-wvv5s 1/1 Running 0 7m50s
```

3. Ejecute el siguiente comando para comprobar si se han modificado los archivos de la ruta **/data** del nuevo pod:

```
kubectl exec web-demo-846b489584-d4d4j -- ls /data
```

Producto esperado:

```
static
```

Si el archivo **static** todavía existe, los datos se pueden almacenar de forma persistente.

### Paso 5 Verifique el uso compartido de datos.

1. Ejecute el siguiente comando para ver el pod creado:

```
kubectl get pod | grep web-demo
```

Producto esperado:

```
web-demo-846b489584-d4d4j 1/1 Running 0 7m
web-demo-846b489584-wvv5s 1/1 Running 0 13m
```

2. Ejecute el siguiente comando para crear un archivo llamado **share** en la ruta **/data** de cualquier pod: En este ejemplo, seleccione el pod llamado **web-demo-846b489584-d4d4j**.

```
kubectl exec web-demo-846b489584-d4d4j -- touch /data/share
```

Compruebe los archivos en la ruta **/data** del pod.

```
kubectl exec web-demo-846b489584-d4d4j -- ls /data
```

Producto esperado:

```
share
static
```

3. Compruebe si el archivo **share** existe en la ruta **/data** de otro pod (**web-demo-846b489584-wvv5s**) también para verificar el uso compartido de datos.

```
kubectl exec web-demo-846b489584-wvv5s -- ls /data
```

Producto esperado:

```
share
static
```

Después de crear un archivo en la ruta **/data** de un pod, si el archivo también se crea en la ruta **/data** de otros pods, los dos pods comparten el mismo volumen.

---Fin

## Operaciones relacionadas

También puede realizar las operaciones que aparecen en [Tabla 8-35](#).

**Tabla 8-35** Operaciones relacionadas

Operación	Descripción	Procedimiento
Actualización de una clave de acceso	Actualice la clave de acceso del almacenamiento de objetos en la consola de CCE.	<ol style="list-style-type: none"> <li>1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumeClaims (PVCs)</b>. Haga clic en <b>More &gt; Update Access Key</b> en la columna <b>Operation</b> del PVC.</li> <li>2. Sube un archivo clave en formato .csv. Para obtener más información, véase <b>Obtención de una clave de acceso</b>. Haga clic en <b>OK</b>.</li> </ol> <p><b>NOTA</b>                      Después de actualizar una clave de acceso global, se puede acceder a todos los pods montados con el almacenamiento de objetos que utiliza esta clave de acceso solo después de reiniciarse.</p>
Eventos	Puede ver los nombres de eventos, los tipos de eventos, el número de ocurrencias, los eventos de Kubernetes, la hora de primera ocurrencia y la hora de última ocurrencia del PVC o PV.	<ol style="list-style-type: none"> <li>1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumeClaims (PVCs)</b> o <b>PersistentVolumes (PVs)</b>.</li> <li>2. Haga clic en <b>View Events</b> en la columna <b>Operation</b> del PVC o del PV de destino para ver los eventos generados en una hora (los datos de eventos se conservan durante una hora).</li> </ol>
Consulta de un archivo YAML	Puede ver, copiar y descargar los archivos YAML de un PVC o PV.	<ol style="list-style-type: none"> <li>1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumeClaims (PVCs)</b> o <b>PersistentVolumes (PVs)</b>.</li> <li>2. Haga clic en <b>View YAML</b> en la columna <b>Operation</b> del PVC o del PV de destino para ver o descargar el YAML.</li> </ol>

## 8.6.4 Configuración de opciones de montaje de OBS

Esta sección describe cómo configurar las opciones de montaje de volumen de OBS. Puede configurar las opciones de montaje en un PV y vincularlo a un PVC. Alternativamente, configure las opciones de montaje en un StorageClass y use el StorageClass para crear un PVC. De esta manera, los PV se pueden crear dinámicamente y heredar opciones de montaje configuradas en el StorageClass de forma predeterminada.

### Requisitos previos

La versión del complemento más antiguo debe ser **1.2.8 o posterior**. El complemento identifica las opciones de montaje y las transfiere a los recursos de almacenamiento subyacentes, que determinan si las opciones especificadas son válidas.

### Restricciones

Las opciones de montaje no se pueden configurar para el contenedor de Kata.

## Opciones de montaje de OBS

Al montar un volumen de OBS, el complemento siempre preestablece las opciones descritas en [Tabla 8-36](#) y [Tabla 8-37](#) de forma predeterminada. Las opciones de [Tabla 8-36](#) son obligatorias. Puede configurar otras opciones de montaje si es necesario. Para obtener más información, consulte [Montaje de un sistema de archivos paralelo](#).

**Tabla 8-36** Opciones de montaje obligatorias configuradas por defecto

Parámetro	Descripción
use_ino	Si está habilitado, obsfs asigna el número <b>inode</b> . Habilitado por defecto en el modo de lectura/escritura.
big_writes	Si se configura, se puede modificar el tamaño máximo de la caché.
nonempty	Permite rutas de montaje no vacías.
allow_other	Permite a otros usuarios acceder al sistema de archivos paralelo.
no_check_certificate	Deshabilita la verificación de certificados del servidor.
enable_noobj_cache	Habilita las entradas de caché para objetos que no existen, lo que puede mejorar el rendimiento. Habilitado por defecto en el modo de lectura/escritura del bucket de objetos.  <b>Esta opción ya no está establecida de forma predeterminada desde everest 1.2.40.</b>
sigv2	Especifica la versión de firma. Se utiliza de forma predeterminada en bucket de objetos.

**Tabla 8-37** Opciones de montaje opcionales configuradas por defecto

Parámetro	Descripción
max_write=131072	Este parámetro solo es válido cuando <b>big_writes</b> está configurado. El valor recomendado es <b>128 KB</b> .
ssl_verify_hostname=0	Deshabilita la verificación del certificado SSL basada en el nombre del host.
max_background=100	Permite establecer el número máximo de solicitudes de espera en segundo plano. Se utiliza de forma predeterminada en sistemas de archivos paralelos.
public_bucket=1	Si se establece en <b>1</b> , los bucket públicos se montan de forma anónima. Habilitado por defecto en el modo de lectura/escritura del bucket de objetos.
umask	Máscara del permiso del archivo de configuración.

## Configuración de opciones de montaje en un PV

Puede utilizar el campo **mountOptions** para configurar las opciones de montaje en un PV. Las opciones que puede configurar en **mountOptions** se enumeran en [Opciones de montaje de OBS](#).

**Paso 1** Utilice `kubectl` para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).

**Paso 2** Configure las opciones de montaje en un PV. Por ejemplo:

```
apiVersion: v1
kind: PersistentVolume
metadata:
  annotations:
    pv.kubernetes.io/provisioned-by: everest-csi-provisioner
    everest.io/reclaim-policy: retain-volume-only # (Optional) The PV is
    deleted while the underlying volume is retained.
  name: pv-obs # PV name.
spec:
  accessModes:
    - ReadWriteMany # Access mode. The value must be ReadWriteMany for OBS.
  capacity:
    storage: 1Gi # OBS volume capacity.
  csi:
    driver: obs.csi.everest.io # Dependent storage driver for the mounting.
    fsType: obsfs # Instance type.
    volumeHandle: <your_volume_id> # Name of the OBS volume.
    volumeAttributes:
      storage.kubernetes.io/csiProvisionerIdentity: everest-csi-provisioner
      everest.io/obs-volume-type: STANDARD
      everest.io/region: <your_region> # Region where the
      OBS volume is.
      everest.io/enterprise-project-id: <your_project_id> # (Optional)
      Enterprise project ID. If an enterprise project is specified, you need to use the
      same enterprise project when creating a PVC. Otherwise, the PVC cannot be bound
      to a PV.
    nodePublishSecretRef: # Custom secret of the OBS volume.
      name: <your_secret_name> # Custom secret name.
      namespace: <your_namespace> # Namespace of the custom secret.
  persistentVolumeReclaimPolicy: Retain # Reclaim policy.
  storageClassName: csi-obs # Storage class name.
  mountOptions: # Mount options.
    - umask=0027
```

**Paso 3** Después de crear un PV, puede crear un PVC y vincularlo al PV y, a continuación, montar el PV en el contenedor en la carga de trabajo. Para obtener más información, véase [Uso de un bucket de OBS existente con un PV estático](#).

**Paso 4** Compruebe si las opciones de montaje tienen efecto.

En este ejemplo, el PVC se monta en la carga de trabajo que utiliza la imagen **nginx:latest**. Puede iniciar sesión en el nodo donde reside el pod en el que está montado el volumen OBS y ver los detalles de progreso.

Ejecute el siguiente comando:

- Bucket de objetos **ps -ef | grep s3fs**

```
root      22142      1   0 Jun03 ?           00:00:00 /usr/bin/s3fs
{your_obs_name} /mnt/paas/kubernetes/kubelet/pods/{pod_uid}/volumes/
kubernetes.io~csi/{your_pv_name}/mount -o url=https://{endpoint}:443 -o
endpoint={region} -o passwd_file=/opt/everest-host-connector/***_obstmpcred/
{your_obs_name} -o nonempty -o big_writes -o sigv2 -o allow_other -o
no_check_certificate -o ssl_verify_hostname=0 -o umask=0027 -o
max_write=131072 -o multipart_size=20
```

- Sistema de archivos paralelo **ps -ef | grep obsfs**

```
root      1355      1  0 Jun03 ?          00:03:16 /usr/bin/obsfs
{your_obs_name} /mnt/paas/kubernetes/kubelet/pods/{pod_uid}/volumes/
kubernetes.io~csi/{your_pv_name}/mount -o url=https://{endpoint}:443 -o
endpoint={region} -o passwd_file=/opt/everest-host-connector/***_obstmpcred/
{your_obs_name} -o allow_other -o nonempty -o big_writes -o use_ino -o
no_check_certificate -o ssl_verify_hostname=0 -o max_background=100 -o
umask=0027 -o max_write=131072
```

----Fin

## Configuración de las opciones de montaje en un StorageClass

Puede utilizar el campo **mountOptions** para configurar las opciones de montaje en un StorageClass. Las opciones que puede configurar en **mountOptions** se enumeran en **Opciones de montaje de OBS**.

**Paso 1** Utilice kubectl para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).

**Paso 2** Cree un StorageClass personalizado. Por ejemplo:

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: csi-obs-mount-option
provisioner: everest-csi-provisioner
parameters:
  csi.storage.k8s.io/csi-driver-name: obs.csi.everest.io
  csi.storage.k8s.io/fstype: s3fs
  everest.io/obs-volume-type: STANDARD
reclaimPolicy: Delete
volumeBindingMode: Immediate
mountOptions:
  - umask=0027 # Mount options.
```

**Paso 3** Una vez configurado el StorageClass, puede usarlo para crear un PVC. De forma predeterminada, los PV creados dinámicamente heredan las opciones de montaje configuradas en el StorageClass. Para obtener más información, véase [Uso de un bucket de OBS con un PV dinámico](#).

**Paso 4** Compruebe si las opciones de montaje tienen efecto.

En este ejemplo, el PVC se monta en la carga de trabajo que utiliza la imagen **nginx:latest**. Puede iniciar sesión en el nodo donde reside el pod en el que está montado el volumen OBS y ver los detalles de progreso.

Ejecute el siguiente comando:

- Bucket de objetos **ps -ef | grep s3fs**

```
root      22142     1  0 Jun03 ?          00:00:00 /usr/bin/s3fs
{your_obs_name} /mnt/paas/kubernetes/kubelet/pods/{pod_uid}/volumes/
kubernetes.io~csi/{your_pv_name}/mount -o url=https://{endpoint}:443 -o
endpoint={region} -o passwd_file=/opt/everest-host-connector/***_obstmpcred/
{your_obs_name} -o nonempty -o big_writes -o sigv2 -o allow_other -o
no_check_certificate -o ssl_verify_hostname=0 -o umask=0027 -o
max_write=131072 -o multipart_size=20
```

- Sistema de archivos paralelo **ps -ef | grep obsfs**

```
root      1355      1  0 Jun03 ?          00:03:16 /usr/bin/obsfs
{your_obs_name} /mnt/paas/kubernetes/kubelet/pods/{pod_uid}/volumes/
kubernetes.io~csi/{your_pv_name}/mount -o url=https://{endpoint}:443 -o
endpoint={region} -o passwd_file=/opt/everest-host-connector/***_obstmpcred/
{your_obs_name} -o allow_other -o nonempty -o big_writes -o use_ino -o
no_check_certificate -o ssl_verify_hostname=0 -o max_background=100 -o
umask=0027 -o max_write=131072
```

----Fin

## 8.6.5 Uso de una AK/SK personalizada para montar un volumen de OBS

### Escenario

Puede resolver este problema con Everest 1.2.8 y las versiones posteriores para usar claves de acceso personalizadas para diferentes usuarios de IAM. Para obtener más información, vea [¿Cómo puedo controlar los permisos de acceso para OBS?](#)

### Requisitos previos

- La versión del complemento más antiguo debe ser 1.2.8 o posterior.
- La versión del clúster debe ser 1.15.11 o posterior.

### Restricciones

Las claves de acceso personalizadas no se pueden configurar para los contenedores seguros.

### Desactivación del montaje automático de la llave

La clave que ha cargado se utiliza de forma predeterminada al montar un volumen de OBS. Es decir, todos los usuarios de IAM bajo su cuenta usarán la misma clave para montar bucket de OBS, y tienen los mismos permisos en bucket. Esta configuración no permite configurar permisos diferenciados para diferentes usuarios de IAM.

Si ha cargado la AK/SK, se recomienda desactivar el montaje automático de claves de acceso activando el parámetro **disable\_auto\_mount\_secret** en el complemento everest para evitar que los usuarios de IAM realicen operaciones no autorizadas. De esta manera, las claves de acceso cargadas en la consola no se usarán al crear volúmenes de OBS.

#### NOTA

- Al habilitar **disable-auto-mount-secret**, asegúrese de que no existe ningún volumen de OBS en el clúster. Una carga de trabajo montada con un volumen de OBS, cuando se escala o se reinicia, no podrá volver a montar el volumen de OBS porque necesita especificar la clave de acceso, pero **disable-auto-mount-secret** lo prohíbe.
- Si **disable-auto-mount-secret** se establece en **true**, se debe especificar una clave de acceso cuando se crea un PV o PVC. De lo contrario, el volumen de OBS no se puede montar.

#### kubectl edit ds everest-csi-driver -nkube-system

Busque **disable-auto-mount-secret** y configúrelo en **true**.

```

- /bin/sh
- -c
- /var/paas/everest-csi-driver/everest-csi-driver --call_node=kubelst --drivers=* --local_csi_everest.io
--aksk-secret-name=paas.aksk --iam-endpoint=https://iam.cn-north-7.ulangab.huawei.com:443 --cvs-endpoint=https://cvs.cn-north-7.ulangab.huawei.com:443
--ecs-endpoint=https://ecs.cn-north-7.ulangab.huawei.com:443 --sfs-endpoint=https://sfs.cn-north-7.ulangab.huawei.com:443
--obs-endpoint=https://obs.cn-north-7.ulangab.huawei.com:443 --sfs-turbo-endpoint=https://sfs-turbo.cn-north-7.myhuaweicloud.com:443
--bms-endpoint=https://bms.cn-north-7.ulangab.huawei.com:443 --ims-endpoint=https://ims.cn-north-7.ulangab.huawei.com:443
--feature-gates=supporthks=false --project-id=b6315dd3d0ff4be5b31a963256794989
--cluster-id=827dced9-c2ad-11e9-bfce-0255ac1036e0 --default-vpc-id=0f090290-2b77-48ae-a601-0e746f350265
+disable-auto-mount-secret=true --cluster-version=v1.19.10-r0 --v=2 1>>/var/paas/sys/log/everest-csi-driver/everest-csi-driver-standalone.log
    
```

Ejecute **:wq** para guardar la configuración y salir. Espere hasta que se reinicie el pod.

### Obtención de una clave de acceso

**Paso 1** Inicie sesión en la consola.

- Paso 2** Pase el cursor sobre el nombre de usuario en la esquina superior derecha y elija **My Credentials** en la lista desplegable.
  - Paso 3** En el panel de navegación, elija **Access Keys**.
  - Paso 4** Haga clic en **Create Access Key**. Aparece el cuadro de diálogo **Create Access Key**.
  - Paso 5** Haga clic en **OK** para descargar la clave de acceso.
- Fin

## Creación de un secreto mediante una clave de acceso

- Paso 1** Obtenga una clave de acceso.
- Paso 2** Codifique las claves con Base64. (Asume que la AK es xxx y la SK es yyyy.)

```
echo -n xxx|base64
```

```
echo -n yyy|base64
```

Registre las AK y SK codificadas.

- Paso 3** Cree un archivo YAML para el secreto, por ejemplo, **test-user.yaml**.

```
apiVersion: v1
data:
  access.key: WE5WWVhVNU*****
  secret.key: Nnk4emJyZ0*****
kind: Secret
metadata:
  name: test-user
  namespace: default
  labels:
    secret.kubernetes.io/used-by: csi
type: cfe/secure-opaque
```

En especial:

Parámetro	Descripción
access.key	AK codificada en Base64.
secret.key	SK codificada en Base64.
name	Nombre de secreto.
namespace	Espacio de nombres del secreto.
secret.kubernetes.io/used-by: csi	Necesita agregar esta etiqueta en el archivo YAML si desea que esté disponible en la consola de CCE cuando cree un PV/PVC de OBS.
type	Tipo secreto. El valor debe ser <b>cfesecureopaque</b> . Cuando se utiliza este tipo, los datos introducidos por los usuarios se cifran automáticamente.

- Paso 4** Cree el secreto.

**kubectl create -f test-user.yaml**

----Fin

## Montar un secreto al crear estáticamente un volumen de OBS

Después de crear un secreto con la AK/SK, puede asociar el secreto con el PV a crear y luego usar la AK/SK en secreto para montar un volumen de OBS.

**Paso 1** Inicie sesión en la consola de OBS, cree un bucket de OBS y registre el nombre del bucket y la clase de almacenamiento. El sistema de archivos paralelo se utiliza como ejemplo.

**Paso 2** Cree un archivo YAML para el PV, por ejemplo, **pv-example.yaml**.

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-obs-example
  annotations:
    pv.kubernetes.io/provisioned-by: everest-csi-provisioner
spec:
  accessModes:
    - ReadWriteMany
  capacity:
    storage: 1Gi
  csi:
    nodePublishSecretRef:
      name: test-user
      namespace: default
    driver: obs.csi.everest.io
    fsType: obsfs
    volumeAttributes:
      everest.io/obs-volume-type: STANDARD
      everest.io/region: ap-southeast-1
      storage.kubernetes.io/csiProvisionerIdentity: everest-csi-provisioner
    volumeHandle: obs-normal-static-pv
    persistentVolumeReclaimPolicy: Delete
    storageClassName: csi-obs
```

Parámetro	Descripción
nodePublishSecretRef	Secreto especificado durante el montaje. <ul style="list-style-type: none"> <li>● <b>name</b>: nombre del secreto</li> <li>● <b>namespace</b>: espacio de nombres del secreto</li> </ul>
fsType	Tipo de archivo. El valor puede ser <b>obsfs</b> o <b>s3fs</b> . Si el valor es de <b>s3fs</b> , se crea un bucket de OBS y se monta usando s3fs. Si el valor es de <b>obsfs</b> , se crea un sistema de archivos paralelo de OBS y se monta usando obsfs. Se recomienda establecer este campo en <b>obsfs</b> .
volumeHandle	Nombre del bucket de OBS.

**Paso 3** Cree un PV.

**kubectl create -f pv-example.yaml**

Después de crear un PV, puede crear un PVC y asociarlo con el PV.

**Paso 4** Cree un archivo YAML para el PVC, por ejemplo, **pvc-example.yaml**.

### Ejemplo de archivo YAML para el PVC:

```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    csi.storage.k8s.io/node-publish-secret-name: test-user
    csi.storage.k8s.io/node-publish-secret-namespace: default
    volume.beta.kubernetes.io/storage-provisioner: everest-csi-provisioner
    everest.io/obs-volume-type: STANDARD
    csi.storage.k8s.io/fstype: obsfs
  name: obs-secret
  namespace: default
spec:
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: csi-obs
  volumeName: pv-obs-example
    
```

Parámetro	Descripción
csi.storage.k8s.io/node-publish-secret-name	Nombre del secreto
csi.storage.k8s.io/node-publish-secret-namespace	Espacio de nombres del secreto

**Paso 5** Cree un PVC.

**kubectl create -f pvc-example.yaml**

Una vez creado el PVC, puede crear una carga de trabajo y asociarla con el PVC para crear volúmenes.

---Fin

## Montar un secreto al crear dinámicamente un volumen de OBS

Al crear dinámicamente un volumen de OBS, puede utilizar el siguiente método para especificar un secreto:

**Paso 1** Cree un archivo YAML para el PVC, por ejemplo, **pvc-example.yaml**.

```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    csi.storage.k8s.io/node-publish-secret-name: test-user
    csi.storage.k8s.io/node-publish-secret-namespace: default
    everest.io/obs-volume-type: STANDARD
    csi.storage.k8s.io/fstype: obsfs
  name: obs-secret
  namespace: default
spec:
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: csi-obs
    
```

Parámetro	Descripción
csi.storage.k8s.io/node-publish-secret-name	Nombre del secreto
csi.storage.k8s.io/node-publish-secret-namespace	Espacio de nombres del secreto

**Paso 2** Cree un PVC.

**kubectl create -f pvc-example.yaml**

Una vez creado el PVC, puede crear una carga de trabajo y asociarla con el PVC para crear volúmenes.

---Fin

**Verificación**

Puede utilizar un secreto de un usuario de IAM para montar un volumen de OBS. Supongamos que se crea una carga de trabajo llamada **obs-secret**, la ruta de montaje en el contenedor es de **/temp** y el usuario de IAM tiene los permisos de CCE **ReadOnlyAccess** y **Tenant Guest**.

1. Consulte el nombre del pod de carga de trabajo.

**kubectl get po | grep obs-secret**

Productos previstos:

```
obs-secret-5cd558f76f-vxslv      1/1      Running    0          3m22s
```

2. Consulte los objetos en la ruta de montaje. En este ejemplo, la consulta se realiza correctamente.

**kubectl exec obs-secret-5cd558f76f-vxslv -- ls -l /temp/**

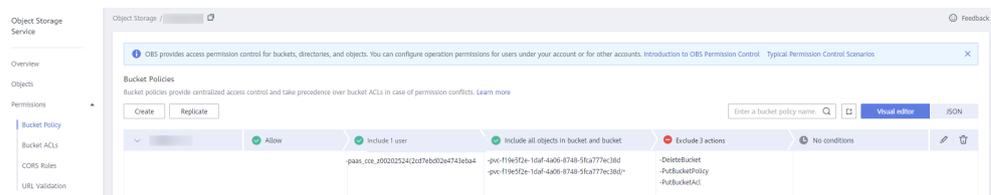
3. Escriba los datos en la ruta de montaje. En este ejemplo, la operación de escritura falló.

**kubectl exec obs-secret-5cd558f76f-vxslv -- touch /temp/test**

Productos previstos:

```
touch: setting times of '/temp/test': No such file or directory
command terminated with exit code 1
```

4. Establezca los permisos de lectura/escritura para el usuario de IAM que montó el volumen de OBS haciendo referencia a la configuración de la política de bucket.



5. Escriba los datos en el camino de la boca de nuevo. En este ejemplo, la operación de escritura se realizó correctamente.

**kubectl exec obs-secret-5cd558f76f-vxslv -- touch /temp/test**

6. Compruebe la ruta de montaje en el contenedor para ver si los datos se escriben correctamente.

```
kubectl exec obs-secret-5cd558f76f-vxslv -- ls -l /temp/
```

Productos previstos:

```
-rwxrwxrwx 1 root root 0 Jun 7 01:52 test
```

## 8.6.6 Uso de bucket de OBS en todas las regiones

De forma predeterminada, un pod puede usar bucket de OBS solo en la misma región. CCE permite que una carga de trabajo utilice bucket de OBS en todas las regiones, lo que puede mejorar la utilización de recursos en algunos escenarios, pero también puede dar como resultado una mayor latencia.

### Restricciones

- OBS solo se puede utilizar en regiones en clústeres de CCE de v1.15 y v1.19.
- La versión del complemento más antiguo debe ser **1.2.32 o posterior**.
- El nodo en el que se monta el almacenamiento debe poder acceder a los bucket de OBS. En general, Internet o Direct Connect se utilizan para acceder a los bucket de OBS en todas las regiones. Puede hacer ping al punto de conexión de OBS en el nodo donde se encuentra OBS para comprobar si OBS es accesible.
- Solo los PV pueden usar buckets de OBS en todas las regiones y, a continuación, se enlazan a PVC. La política de recuperación de energía de PV debe ser **Retain**. Las clases de almacenamiento no se pueden utilizar para crear de forma dinámica PVC para utilizar bucket de OBS en todas las regiones.

### Procedimiento

**Paso 1** Cree el ConfigMap de `paas-obs-endpoint` y configure la región y el punto de conexión de OBS.

El nombre de ConfigMap se fija a `paas-obs-endpoint` y el espacio de nombres se fija a `kube-system`.

La región y el punto de conexión se establecen en el formato de pares clave-valor. Reemplace `<region_name>` y `<endpoint_address>` por valores específicos. For details, see [Regiones y puntos de conexión](#). Utilice comas (,) para separar varios valores.

Ejemplo: `{"ap-southeast-1": "https://obs.ap-southeast-1.myhuaweicloud.com:443", "ap-southeast-3": "https://obs.ap-southeast-3.myhuaweicloud.com:443"}`

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: paas-obs-endpoint # The value must be paas-obs-endpoint.
  namespace: kube-system # The value must be kube-system.
data:
  obs-endpoint: |
    {"<region_name>": "<endpoint_address>"}
```

**Paso 2** Cree un PV.

Establezca `everest.io/region` en la región donde se encuentra OBS.

```
kind: PersistentVolume
apiVersion: v1
metadata:
  name: testing-abc
annotations:
  pv.kubernetes.io/bound-by-controller: 'yes'
  pv.kubernetes.io/provisioned-by: everest-csi-provisioner
```

```
spec:
  capacity:
    storage: 1Gi
  csi:
    driver: obs.csi.everest.io
    volumeHandle: testing-abc           # OBS bucket name
    fsType: s3fs                       # obsfs indicates to create a parallel
file system (recommended), and s3fs indicates to create an object bucket.
  volumeAttributes:
    everest.io/obs-volume-type: STANDARD
    everest.io/region: <region_name>   # Region where the OBS bucket
resides. Replace it with a specific value.
    storage.kubernetes.io/csiProvisionerIdentity: everest-csi-provisioner
  nodePublishSecretRef:
    name: test-user                    # AK/SK used for mounting an OBS bucket
    namespace: default
  accessModes:
    - ReadWriteMany
  persistentVolumeReclaimPolicy: Retain # The value must be Retain.
  storageClassName: csi-obs
  volumeMode: Filesystem
```

**nodePublishSecretRef** es la clave de acceso (AK/SK) utilizada para montar el volumen de almacenamiento de objetos. Necesita usar la AK/SK para crear un secreto, que se usará al crear un PV. Para obtener más información, véase [Uso de una AK/SK personalizada para montar un volumen de OBS](#).

### Paso 3 Cree un PVC.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-test-abc
  namespace: default
  annotations:
    everest.io/obs-volume-type: STANDARD # OBS bucket
type. Currently, standard (STANDARD) and infrequent access (WARM) are supported.
    csi.storage.k8s.io/fstype: s3fs     # File type.
obsfs indicates to create a parallel file system (recommended), and s3fs
indicates to create an OBS bucket.
    volume.beta.kubernetes.io/storage-provisioner: everest-csi-provisioner
spec:
  accessModes:
    - ReadWriteMany # For object storage, the value must be
ReadWriteMany.
  resources:
    requests:
      storage: 1Gi # Storage capacity of a PVC. This field is valid
only for verification (fixed to 1, cannot be empty or 0). The value setting does
not take effect for OBS buckets.
      storageClassName: csi-obs # Storage class name. For object storage, the value
is fixed to csi-obs.
      volumeName: testing-abc # PV name
```

**Paso 4** Cree una carga de trabajo, seleccione el PVC en la opción de almacenamiento de datos de la configuración de contenedor y agregue el PVC creado. Si la carga de trabajo se crea correctamente, el bucket de OBS se puede utilizar en todas las regiones.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: obs-deployment-example # Workload name
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      app: obs-deployment-example
  template:
```

```
metadata:
  labels:
    app: obs-deployment-example
spec:
  containers:
  - image: nginx
    name: container-0
    volumeMounts:
    - mountPath: /tmp                # Mount path
      name: pvc-obs-example
  restartPolicy: Always
  imagePullSecrets:
  - name: default-secret
  volumes:
  - name: pvc-obs-example
    persistentVolumeClaim:
      claimName: pvc-test-abc        # PVC name
```

----Fin

## 8.7 PV local

### 8.7.1 Descripción general

#### Presentación

CCE le permite usar LVM para combinar volúmenes de datos en nodos en un grupo de almacenamiento (VolumeGroup) y crear LV para que contenedores los monte. Un PV que utiliza un volumen persistente local como medio se considera el PV local.

En comparación con el volumen de HostPath, el PV local se puede usar de una manera persistente y portátil. Además, el PV del PV local tiene la configuración de afinidad de nodo. El pod montado en el PV local se programa automáticamente en función de la configuración de afinidad. No es necesario programar manualmente el pod a un nodo específico.

#### Modos de montaje

Los PV locales solo se pueden montar en los siguientes modos:

- **Uso de un PV local a través de un PV dinámico:** modo de creación dinámica, donde se especifica un StorageClass durante la creación de PVC y se creará automáticamente un volumen de OBS y un PV.
- **Montaje dinámico de un PV local en un StatefulSet:** Solo StatefulSets admite este modo. Cada pod está asociado con un PVC y PV únicos. Después de reprogramar un pod, los datos originales todavía se pueden montar en él basándose en el nombre de PVC. Este modo es aplicable a StatefulSets con múltiples pods.

#### Restricciones

- Los PV locales solo se admiten cuando la versión del clúster es v1.21.2-r0 o posterior y la versión adicional más reciente es 2.1.23 o posterior. Se recomienda 2.1.23 o posterior.
- Quitar, eliminar, restablecer o ajustar en un nodo hará que se pierdan los datos de PVC/PV del PV local asociado con el nodo, que no se pueden restaurar o usar de nuevo. Para obtener más información, consulte [Quitar un nodo](#), [Eliminar un nodo](#), [Restablecer un nodo](#) y [Ajustar un nodo](#). En estos escenarios, el pod que utiliza el PV local es desalojado del nodo. Se creará un nuevo pod y permanecerá en el estado

pendiente. Esto se debe a que el PVC utilizado por el pod tiene una etiqueta de nodo, debido a lo cual el pod no se puede programar. Después de que se restablezca el nodo, el pod puede planificarse para el nodo de reinicio. En este caso, el pod permanece en el estado de creación porque el volumen lógico subyacente correspondiente al PVC no existe.

- No elimine manualmente el grupo de almacenamiento correspondiente ni desconecte los discos de datos del nodo. De lo contrario, pueden producirse excepciones como la pérdida de datos.
- Un PV local no se puede montar en múltiples cargas de trabajo o trabajos al mismo tiempo.

## 8.7.2 Uso de un PV local a través de un PV dinámico

### Requisitos previos

- Ha creado un clúster e instalado el complemento de CSI ([everest](#)) en el clúster.
- Si desea crear un clúster mediante comandos, utilice `kubectl` para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).
- Ha importado un disco de datos de un nodo al grupo de almacenamiento de PV local. Para obtener más información, véase [Grupos de almacenamiento](#).

### Restricciones

- Los PV locales solo se admiten cuando la versión del clúster es v1.21.2-r0 o posterior y la versión adicional más reciente es 2.1.23 o posterior. Se recomienda 2.1.23 o posterior.
- Quitar, eliminar, restablecer o ajustar en un nodo hará que se pierdan los datos de PVC/PV del PV local asociado con el nodo, que no se pueden restaurar o usar de nuevo. Para obtener más información, consulte [Quitar un nodo](#), [Eliminar un nodo](#), [Restablecer un nodo](#) y [Ajustar un nodo](#). En estos escenarios, el pod que utiliza el PV local es desalojado del nodo. Se creará un nuevo pod y permanecerá en el estado pendiente. Esto se debe a que el PVC utilizado por el pod tiene una etiqueta de nodo, debido a lo cual el pod no se puede programar. Después de que se restablezca el nodo, el pod puede planificarse para el nodo de reinicio. En este caso, el pod permanece en el estado de creación porque el volumen lógico subyacente correspondiente al PVC no existe.
- No elimine manualmente el grupo de almacenamiento correspondiente ni desconecte los discos de datos del nodo. De lo contrario, pueden producirse excepciones como la pérdida de datos.
- Un PV local no se puede montar en múltiples cargas de trabajo o trabajos al mismo tiempo.

### Creación automática de un PV local en la consola

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** Cree dinámicamente un PVC y un PV.

1. Elija **Storage** en el panel de navegación y haga clic en la ficha **PersistentVolumeClaims (PVCs)**. Haga clic en **Create PVC** en la esquina superior derecha. En el cuadro de diálogo que se muestra, configure los parámetros de PVC.

Parámetro	Descripción
PVC Type	En esta sección, seleccione <b>Local PV</b> .
PVC Name	Escriba el nombre de PVC, que debe ser único en el mismo espacio de nombres.
Creation Method	Solo puede seleccionar <b>Dynamically provision</b> para crear un PVC, PV y almacenamiento subyacente en la consola en modo en cascada.
Storage Classes	La clase de almacenamiento de los PV locales es de <b>csi-local-topology</b> .
Access Mode	Los PV locales solo soportan <b>ReadWriteOnce</b> , lo que indica que un volumen de almacenamiento puede montarse en un nodo en modo de lectura/escritura. Para obtener más información, véase <a href="#">Modos de acceso al volumen</a> .
Storage Pool	Ver el grupo de almacenamiento importado. Para obtener más información acerca de cómo importar un nuevo volumen de datos al grupo de almacenamiento, consulte <a href="#">Grupos de almacenamiento</a> .
Capacity (GiB)	Capacidad del volumen de almacenamiento solicitado.

- Haga clic en **Create** para crear un PVC y un PV.

Puede elegir **Storage** en el panel de navegación y ver el PVC y PV creados en las páginas de fichas **PersistentVolumeClaims (PVCs)** y **PersistentVolumes (PVs)**.

 **NOTA**

El modo de enlace de volumen de la clase de almacenamiento local (denominada **csi-local-topology**) es enlace tardío (es decir, el valor de **volumeBindingMode** es **WaitForFirstConsumer**). En este modo, la creación y el enlace de PV se retrasan. El PV correspondiente se crea y enlaza solo cuando se utiliza el PVC durante la creación de la carga de trabajo.

**Paso 3** Cree una aplicación.

- En el panel de navegación de la izquierda, haga clic en **Workloads**. En el panel derecho, haga clic en la ficha **Deployments**.
- Haga clic en **Create Workload** en la esquina superior derecha. En la página mostrada, haga clic en **Data Storage** en el área **Container Settings** y haga clic en **Add Volume** para seleccionar **PVC**.

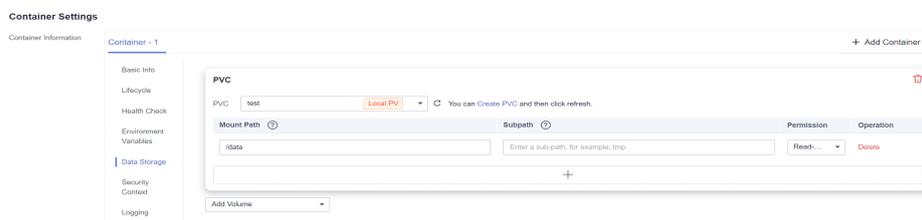
Montar y utilizar volúmenes de almacenamiento, como se muestra en [Tabla 8-38](#). Para obtener más información sobre otros parámetros, consulte [Workloads](#).

**Tabla 8-38** Montaje de un volumen de almacenamiento

Parámetro	Descripción
PVC	Seleccione un PV local existente. Un PV local no se puede montar repetidamente en varias cargas de trabajo.

Parámetro	Descripción
Add Container Path	<p>Configure los parámetros siguientes:</p> <ol style="list-style-type: none"> <li><b>Mount Path:</b> Introduzca una ruta de acceso del contenedor. Por ejemplo, <code>/tmp</code>. Este parámetro indica la ruta de contenedor en la que se montará un volumen de datos. No monte el volumen en un directorio del sistema como <code>/</code> o <code>/var/run</code>; esta acción puede causar errores de contenedor. Se recomienda montar el volumen en un directorio vacío. Si el directorio no está vacío, asegúrese de que no hay archivos que afecten al inicio de contenedor. De lo contrario, se reemplazarán los archivos, lo que provocará errores de inicio de contenedor o errores de creación de carga de trabajo.                     <p><b>AVISO</b></p>                     Cuando el contenedor está montado en un directorio de alto riesgo, se recomienda utilizar una cuenta con los permisos mínimos para iniciar el contenedor; de lo contrario, los archivos de alto riesgo en la máquina host podrían estar dañados.</li> <li><b>Subpath:</b> Introduzca una ruta secundaria, por ejemplo, <code>tmp</code>. Se utiliza una ruta secundaria para montar un volumen local de modo que se utilice el mismo volumen de datos en un solo pod. Si este parámetro se deja en blanco, la ruta raíz se utiliza de forma predeterminada.</li> <li><b>Permission</b> <ul style="list-style-type: none"> <li><b>Read-only:</b> Solo puede leer los datos en los volúmenes montados.</li> <li><b>Read/Write:</b> Puede modificar los volúmenes de datos montados en la ruta de acceso. Los datos recién escritos no se migran si se migra el contenedor, lo que puede causar la pérdida de datos.</li> </ul> </li> </ol> <p>Puede hacer clic en <b>+</b> para agregar varias rutas y subrutas.</p>

En este ejemplo, el disco se monta en la trayectoria `/data` del contenedor. Los datos contenedor generados en esta ruta se almacenan en el PV local.



3. Después de completar la configuración, haga clic en **Create**.

Una vez creada la carga de trabajo, puede probar **Verificación de la persistencia y el uso compartido de datos**.

----Fin

## (kubectl) Creación automática de un PV local

**Paso 1** Utilice kubectl para conectarse al clúster.

**Paso 2** Utilice **StorageClass** para crear dinámicamente un PVC y un PV.

1. Cree el archivo **pvc-local.yaml**.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-local
  namespace: default
spec:
  accessModes:
    - ReadWriteOnce          # The local PV must adopt ReadWriteOnce.
  resources:
    requests:
      storage: 10Gi          # Size of the local PV.
      storageClassName: csi-local-topology # StorageClass is local PV.
```

**Tabla 8-39** Parámetros clave

Parámetro	Obligatorio	Descripción
storage	Sí	Capacidad solicitada en el PVC, en Gi.
storageClassName	Sí	Nombre de clase de almacenamiento. El nombre de clase de almacenamiento de PV local es <b>csi-local-topology</b> .

2. Ejecute el siguiente comando para crear un PVC:

```
kubectl apply -f pvc-local.yaml
```

**Paso 3** Cree una aplicación.

1. Cree un archivo denominado **web-demo.yaml**. En este ejemplo, el PV local se monta en la trayectoria **/data**.

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: web-local
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      app: web-local
  serviceName: web-local # Headless Service name.
  template:
    metadata:
      labels:
        app: web-local
    spec:
      containers:
        - name: container-1
          image: nginx:latest
          volumeMounts:
            - name: pvc-disk #Volume name, which must be the same as the
              volume name in the volumes field.
              mountPath: /data #Location where the storage volume is mounted.
      imagePullSecrets:
        - name: default-secret
```

```

volumes:
  - name: pvc-disk      #Volume name, which can be customized.
    persistentVolumeClaim:
      claimName: pvc-local      #Name of the created PVC.
---
apiVersion: v1
kind: Service
metadata:
  name: web-local      # Headless Service name.
  namespace: default
  labels:
    app: web-local
spec:
  selector:
    app: web-local
  clusterIP: None
  ports:
    - name: web-local
      targetPort: 80
      nodePort: 0
      port: 80
      protocol: TCP
  type: ClusterIP
    
```

2. Ejecute el siguiente comando para crear una aplicación en la que está montado el PV local:

```
kubectl apply -f web-local.yaml
```

Una vez creada la carga de trabajo, puede probar [Verificación de la persistencia de datos](#).

----Fin

## Verificación de la persistencia de datos

**Paso 1** Vea la aplicación desplegada y los archivos locales.

1. Ejecute el siguiente comando para ver el pod creado:

```
kubectl get pod | grep web-local
```

Producto esperado:

```
web-local-0          1/1      Running    0          38s
```

2. Ejecute el siguiente comando para comprobar si el PV local se ha montado en la ruta **/data**:

```
kubectl exec web-local-0 -- df | grep data
```

Producto esperado:

```
/dev/mapper/vg--everest--localvolume--persistent-pvc-local
10255636      36888    10202364    0% /data
```

3. Ejecute el siguiente comando para ver los archivos en la ruta **/data**:

```
kubectl exec web-local-0 -- ls /data
```

Producto esperado:

```
lost+found
```

**Paso 2** Ejecute el siguiente comando para crear un archivo llamado **static** en la ruta **/data**:

```
kubectl exec web-local-0 -- touch /data/static
```

**Paso 3** Ejecute el siguiente comando para ver los archivos en la ruta **/data**:

```
kubectl exec web-local-0 -- ls /data
```

Producto esperado:

```
lost+found
static
```

**Paso 4** Ejecute el siguiente comando para eliminar el pod llamado **web-local-0**:

```
kubectl delete pod web-local-0
```

Producto esperado:

```
pod "web-local-0" deleted
```

**Paso 5** Después de la eliminación, el controlador de StatefulSet crea automáticamente una réplica con el mismo nombre. Ejecute el siguiente comando para comprobar si se han modificado los archivos de la ruta **/data**:

```
kubectl exec web-local-0 -- ls /data
```

Producto esperado:

```
lost+found
static
```

Si el archivo **static** todavía existe, los datos en el PV local pueden almacenarse de forma persistente.

---Fin

## Operaciones relacionadas

También puede realizar las operaciones que aparecen en [Tabla 8-40](#).

**Tabla 8-40** Operaciones relacionadas

Operación	Descripción	Procedimiento
Consulta de eventos	Puede ver los nombres de eventos, los tipos de eventos, el número de ocurrencias, los eventos de Kubernetes, la hora de primera ocurrencia y la hora de última ocurrencia del PVC o PV.	<ol style="list-style-type: none"> <li>1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumeClaims (PVCs)</b> o <b>PersistentVolumes (PVs)</b>.</li> <li>2. Haga clic en <b>View Events</b> en la columna <b>Operation</b> del PVC o del PV de destino para ver los eventos generados en una hora (los datos de eventos se conservan durante una hora).</li> </ol>
Consulta de un archivo YAML	Puede ver, copiar y descargar los archivos YAML de un PVC o PV.	<ol style="list-style-type: none"> <li>1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumeClaims (PVCs)</b> o <b>PersistentVolumes (PVs)</b>.</li> <li>2. Haga clic en <b>View YAML</b> en la columna <b>Operation</b> del PVC o del PV de destino para ver o descargar el YAML.</li> </ol>

## 8.7.3 Montaje dinámico de un PV local en un StatefulSet

### Escenarios de aplicación

El montaje dinámico solo está disponible para crear un **StatefulSet**. Se implementa con una plantilla de reclamo de volumen (campo **volumeClaimTemplates**) y depende de la clase de

almacenamiento para aprovisionar PV dinámicamente. En este modo, cada pod en un StatefulSet de múltiples pods está asociado con un PVC y un PV únicos. Después de reprogramar un pod, los datos originales todavía se pueden montar en él basándose en el nombre de PVC. En el modo de montaje común para una Deployment, si se admite ReadWriteMany, varios pods de la Deployment se montarán en el mismo almacenamiento subyacente.

## Requisitos previos

- Ha creado un clúster e instalado el complemento de CSI ([everest](#)) en el clúster.
- Si desea crear un clúster mediante comandos, utilice kubectl para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).
- Ha importado un disco de datos de un nodo al grupo de almacenamiento de PV local. Para obtener más información, véase [Grupos de almacenamiento](#).

## Montaje dinámico de un PV local en la consola

- Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.
- Paso 2** En el panel de navegación de la izquierda, haga clic en **Workloads**. En el panel derecho, haga clic en la ficha **StatefulSets**.
- Paso 3** Haga clic en **Create Workload** en la esquina superior derecha. En la página mostrada, haga clic en **Data Storage** en el área **Container Settings** y haga clic en **Add Volume** para seleccionar **VolumeClaimTemplate (VTC)**.
- Paso 4** Haga clic en **Create PVC**. En el cuadro de diálogo que se muestra, configure los parámetros de plantilla de notificación de volumen.

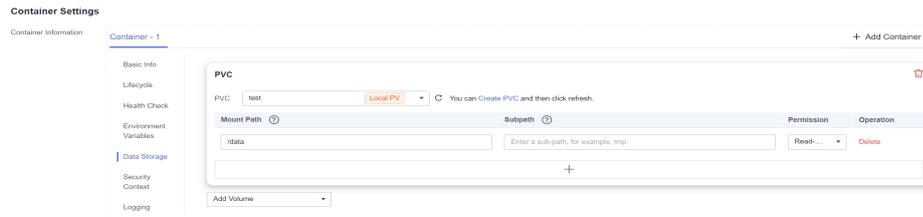
Montar y utilizar dinámicamente volúmenes de almacenamiento. Para obtener más información sobre otros parámetros, consulte [Creación de un StatefulSet](#).

Parámetro	Descripción
PVC Type	En esta sección, seleccione <b>Local PV</b> .
PVC Name	Escriba el nombre del PVC. Después de crear un PVC, se agrega automáticamente un sufijo en función del número de instancias. El formato es <i>&lt;Custom PVC name&gt;-&lt;Serial number&gt;</i> , por ejemplo, <i>example-0</i> .
Creation Method	Solo puede seleccionar <b>Dynamically provision</b> para crear un PVC, PV y almacenamiento subyacente en la consola en modo en cascada.
Storage Classes	La clase de almacenamiento de los PV locales es de <b>csi-local-topology</b> .
Access Mode	Los PV locales solo soportan <b>ReadWriteOnce</b> , lo que indica que un volumen de almacenamiento puede montarse en un nodo en modo de lectura/escritura. Para obtener más información, véase <a href="#">Modos de acceso al volumen</a> .
Storage Pool	Consulte el grupo de almacenamiento importado. Para obtener más información acerca de cómo importar un nuevo volumen de datos al grupo de almacenamiento, consulte <a href="#">Grupos de almacenamiento</a> .

Parámetro	Descripción
Capacity (GiB)	Capacidad del volumen de almacenamiento solicitado.

**Paso 5** Introduzca la ruta en la que está montado el volumen.

En este ejemplo, el disco se monta en la trayectoria **/data** del contenedor. Los datos contenedor generados en esta ruta se almacenan en el PV local.



**Paso 6** Después de completar la configuración, haga clic en **Create**.

Una vez creada la carga de trabajo, puede probar [Verificación de la persistencia de datos](#).

----Fin

## (kubectl) Uso de un PV local existente

**Paso 1** Utilice kubectl para conectarse al clúster.

**Paso 2** Cree un archivo denominado **statefulset-local.yaml**. En este ejemplo, el PV local se monta en la trayectoria **/data**.

```

apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: statefulset-local
  namespace: default
spec:
  selector:
    matchLabels:
      app: statefulset-local
  template:
    metadata:
      labels:
        app: statefulset-local
    spec:
      containers:
        - name: container-1
          image: nginx:latest
          volumeMounts:
            - name: pvc-local # The value must be the same as that in
              # Location where the storage volume is
              mountPath: /data
          imagePullSecrets:
            - name: default-secret
      serviceName: statefulset-local # Headless Service name.
      replicas: 2
      volumeClaimTemplates:
        - apiVersion: v1
          kind: PersistentVolumeClaim
          metadata:
            name: pvc-local
            namespace: default
    
```

```

spec:
  accessModes:
    - ReadWriteOnce          # The local PV must adopt ReadWriteOnce.
  resources:
    requests:
      storage: 10Gi        # Storage volume capacity.
      storageClassName: csi-local-topology # StorageClass is local PV.
---
apiVersion: v1
kind: Service
metadata:
  name: statefulset-local   # Headless Service name.
  namespace: default
  labels:
    app: statefulset-local
spec:
  selector:
    app: statefulset-local
  clusterIP: None
  ports:
    - name: statefulset-local
      targetPort: 80
      nodePort: 0
      port: 80
      protocol: TCP
      type: ClusterIP
    
```

**Tabla 8-41** Parámetros clave

Parámetro	Obligatorio	Descripción
storage	Sí	Capacidad solicitada en el PVC, en Gi.
storageClassName	Sí	La clase de almacenamiento de los PV locales es de <b>csi-local-topology</b> .

**Paso 3** Ejecute el siguiente comando para crear una aplicación en la que está montado el PV local:

```
kubectl apply -f statefulset-local.yaml
```

Una vez creada la carga de trabajo, puede probar [Verificación de la persistencia de datos](#).

----Fin

## Verificación de la persistencia de datos

**Paso 1** Vea la aplicación y los archivos desplegados.

1. Ejecute el siguiente comando para ver el pod creado:

```
kubectl get pod | grep statefulset-local
```

Producto esperado:

```
statefulset-local-0      1/1      Running    0          45s
statefulset-local-1      1/1      Running    0          28s
```

2. Ejecute el siguiente comando para comprobar si el PV local se ha montado en la ruta /**data**:

```
kubectl exec statefulset-local-0 -- df | grep data
```

Producto esperado:

```
/dev/mapper/vg--everest--localvolume--persistent-pvc-local
10255636      36888  10202364    0% /data
```

3. Ejecute el siguiente comando para ver los archivos en la ruta **/data**:

```
kubectl exec statefulset-local-0 -- ls /data
```

Producto esperado:

```
lost+found
```

- Paso 2** Ejecute el siguiente comando para crear un archivo llamado **static** en la ruta **/data**:

```
kubectl exec statefulset-local-0 -- touch /data/static
```

- Paso 3** Ejecute el siguiente comando para ver los archivos en la ruta **/data**:

```
kubectl exec statefulset-local-0 -- ls /data
```

Producto esperado:

```
lost+found  
static
```

- Paso 4** Ejecute el siguiente comando para eliminar el pod llamado **web-local-auto-0**:

```
kubectl delete pod statefulset-local-0
```

Producto esperado:

```
pod "statefulset-local-0" deleted
```

- Paso 5** Después de la eliminación, el controlador de StatefulSet crea automáticamente una réplica con el mismo nombre. Ejecute el siguiente comando para comprobar si se han modificado los archivos de la ruta **/data**:

```
kubectl exec statefulset-local-0 -- ls /data
```

Producto esperado:

```
lost+found  
static
```

Si el archivo **static** todavía existe, los datos en el PV local pueden almacenarse de forma persistente.

----Fin

## Operaciones relacionadas

También puede realizar las operaciones que aparecen en [Tabla 8-42](#).

**Tabla 8-42** Operaciones relacionadas

Operación	Descripción	Procedimiento
Consulta de eventos	Puede ver los nombres de eventos, los tipos de eventos, el número de ocurrencias, los eventos de Kubernetes, la hora de primera ocurrencia y la hora de última ocurrencia del PVC o PV.	<ol style="list-style-type: none"> <li>1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumeClaims (PVCs)</b> o <b>PersistentVolumes (PVs)</b>.</li> <li>2. Haga clic en <b>View Events</b> en la columna <b>Operation</b> del PVC o del PV de destino para ver los eventos generados en una hora (los datos de eventos se conservan durante una hora).</li> </ol>

Operación	Descripción	Procedimiento
Consulta de un archivo YAML	Puede ver, copiar y descargar los archivos YAML de un PVC o PV.	<ol style="list-style-type: none"> <li>1. Elija <b>Storage</b> en el panel de navegación y haga clic en la ficha <b>PersistentVolumeClaims (PVCs)</b> o <b>PersistentVolumes (PVs)</b>.</li> <li>2. Haga clic en <b>View YAML</b> en la columna <b>Operation</b> del PVC o del PV de destino para ver o descargar el YAML.</li> </ol>

## 8.8 Volúmenes efímeros (emptyDir)

### 8.8.1 Descripción general

#### Presentación

Algunas aplicaciones requieren almacenamiento adicional, pero no es importante si los datos siguen disponibles después de un reinicio. Por ejemplo, aunque los servicios de caché están limitados por el tamaño de la memoria, los servicios de caché pueden mover los datos utilizados con poca frecuencia al almacenamiento más lento que la memoria. Como resultado, el rendimiento general no se ve afectado significativamente. Otras aplicaciones requieren datos de solo lectura inyectados como archivos, como datos de configuración o secretos.

**Volúmenes efímeros** (EV) en Kubernetes están diseñados para el escenario anterior. Los EV se crean y eliminan junto con los pods después del ciclo de vida del pod.

Los EV comunes en Kubernetes:

- **emptyDir**: vacía en el arranque del pod, con almacenamiento procedente localmente del directorio básico de kubelet (normalmente el disco raíz) o de la memoria. emptyDir se asigna desde el **EV del nodo**. Si los datos de otras fuentes (como archivos de log o datos de niveles de imágenes) ocupan el almacenamiento temporal, la capacidad de almacenamiento puede ser insuficiente.
- **ConfigMap**: Los datos de Kubernetes del tipo ConfigMap se montan en los pods como volúmenes de datos.
- **Secret**: Los datos de Kubernetes del tipo Secret se montan en los pods como volúmenes de datos.

#### Tipos de emptyDir

CCE proporciona los siguientes tipos de emptyDir:

- **emptyDir**: emptyDir nativo de Kubernetes. Su ciclo de vida es el mismo que el de un pod. La memoria se puede especificar como el medio de almacenamiento. Cuando se elimina el pod, el volumen emptyDir se elimina y sus datos se pierden.
- **EV local**: Los discos de datos locales en un nodo forman un **grupo de almacenamiento** (VolumeGroup) con LVM. Los LV se crean como el medio de almacenamiento de emptyDir y se montan en contenedores. Los LV ofrecen un mejor rendimiento que el medio de almacenamiento predeterminado de emptyDir.

## Restricciones

- Los EV locales solo se admiten cuando la versión del clúster es v1.21.2-r0 o posterior y la versión adicional es 1.2.29 o posterior.
- No elimine manualmente el grupo de almacenamiento correspondiente ni desconecte los discos de datos del nodo. De lo contrario, pueden producirse excepciones como la pérdida de datos.
- Asegúrese de que el directorio `/var/lib/kubelet/pods/` no está montado en el pod del nodo. De lo contrario, el pod, montado con tales volúmenes, puede no ser eliminado.

### 8.8.2 Uso de un EV

Los EV se aplican al almacenamiento temporal de datos, la recuperación ante desastres y el uso compartido de datos en tiempo de ejecución. Se eliminará al eliminar o transferir los pods de carga de trabajo. Esta sección describe cómo usar un EV.

CCE proporciona los siguientes tipos de `emptyDir`:

- **emptyDir**: `emptyDir` nativo de Kubernetes. Su ciclo de vida es el mismo que el de un pod. La memoria se puede especificar como el medio de almacenamiento. Cuando se elimina el pod, el volumen `emptyDir` se elimina y sus datos se pierden.
- **EV local**: Los discos de datos locales en un nodo forman un **grupo de almacenamiento** (VolumeGroup) con LVM. Los LV se crean como el medio de almacenamiento de `emptyDir` y se montan en contenedores. Los LV ofrecen un mejor rendimiento que el medio de almacenamiento predeterminado de `emptyDir`.

## Requisitos previos

- Ha creado un clúster e instalado el complemento de CSI (**everest**) en el clúster.
- Si desea crear un clúster mediante comandos, utilice `kubectl` para conectarse al clúster. Para obtener más información, véase **Conexión a un clúster con `kubectl`**.
- Para usar un EV local, necesita importar un disco de datos de un nodo al grupo de almacenamiento EV local. Para obtener más información, véase **Grupos de almacenamiento**.

## Restricciones

- Los EV locales solo se admiten cuando la versión del clúster es v1.21.2-r0 o posterior y la versión adicional es 1.2.29 o posterior.
- No elimine manualmente el grupo de almacenamiento correspondiente ni desconecte los discos de datos del nodo. De lo contrario, pueden producirse excepciones como la pérdida de datos.
- Asegúrese de que el directorio `/var/lib/kubelet/pods/` no está montado en el pod del nodo. De lo contrario, el pod, montado con tales volúmenes, puede no ser eliminado.

## emptyDir

### Usando la consola

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** En el panel de navegación de la izquierda, haga clic en **Workloads**. En el panel derecho, haga clic en la ficha **Deployments**.

**Paso 3** Haga clic en **Create Workload** en la esquina superior derecha de la página. En el área **Container Settings**, haga clic en la ficha **Data Storage** y haga clic en **Add Volume > emptyDir**.

**Paso 4** Monte y utilice los volúmenes de almacenamiento, como se muestra en [Tabla 8-43](#). Para obtener más información sobre otros parámetros, consulte [Workloads](#).

**Tabla 8-43** Montaje de un EV

Parámetro	Descripción
Storage Medium	<p><b>Memory:</b></p> <ul style="list-style-type: none"> <li>● Puede seleccionar esta opción para mejorar la velocidad de ejecución, pero la capacidad de almacenamiento está sujeta al tamaño de la memoria. Este modo es aplicable cuando el volumen de datos es pequeño y se requiere una lectura y escritura eficientes.</li> <li>● Si esta función está desactivada, los datos se almacenan en discos duros, lo que es aplicable a una gran cantidad de datos con bajos requisitos de eficiencia de lectura y escritura.</li> </ul> <p><b>NOTA</b></p> <ul style="list-style-type: none"> <li>● Si se selecciona <b>Memory</b>, preste atención al tamaño de la memoria. Si la capacidad de almacenamiento excede el tamaño de la memoria, se produce un evento de OOM.</li> <li>● Si se selecciona <b>Memory</b>, el tamaño de un EV es el 50% de las especificaciones del pod y no se puede cambiar.</li> <li>● Si <b>Memory</b> no está seleccionado, los EV no ocuparán la memoria del sistema.</li> </ul>

Parámetro	Descripción
Add Container Path	<p>Configure los parámetros siguientes:</p> <ol style="list-style-type: none"> <li>1. <b>Mount Path:</b> Introduzca una ruta de acceso del contenedor. Por ejemplo, <code>/tmp</code>.                      Este parámetro indica la ruta de contenedor en la que se montará un volumen de datos. No monte el volumen en un directorio del sistema como <code>/</code> o <code>/var/run</code>; esta acción puede causar errores de contenedor. Se recomienda montar el volumen en un directorio vacío. Si el directorio no está vacío, asegúrese de que no hay archivos que afecten al inicio de contenedor. De lo contrario, se reemplazarán los archivos, lo que provocará errores de inicio de contenedor o errores de creación de carga de trabajo.</li> </ol> <p><b>AVISO</b>                      Cuando el contenedor está montado en un directorio de alto riesgo, se recomienda utilizar una cuenta con los permisos mínimos para iniciar el contenedor; de lo contrario, los archivos de alto riesgo en la máquina host podrían estar dañados.</p> <ol style="list-style-type: none"> <li>2. <b>Subpath:</b> Introduzca una ruta secundaria, por ejemplo, <code>tmp</code>.                      Se utiliza una ruta secundaria para montar un volumen local de modo que se utilice el mismo volumen de datos en un solo pod. Si este parámetro se deja en blanco, la ruta raíz se utiliza de forma predeterminada.</li> <li>3. <b>Permission</b> <ul style="list-style-type: none"> <li>– <b>Read-only:</b> Solo puede leer los datos en los volúmenes montados.</li> <li>– <b>Read/Write:</b> Puede modificar los volúmenes de datos montados en la ruta de acceso. Los datos recién escritos no se migran si se migra el contenedor, lo que puede causar la pérdida de datos.</li> </ul> </li> </ol> <p>Puede hacer clic en  para agregar varias rutas y subrutas.</p>

**Paso 5** Después de configurar otros parámetros de carga de trabajo, haga clic en **Create**.

---Fin

### Usando kubectl

**Paso 1** Utilice kubectl para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).

**Paso 2** Cree un archivo llamado `nginx-emptydir.yaml` y edítelo.

**vi nginx-emptydir.yaml**

Contenido del archivo YAML:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-emptydir
  namespace: default
spec:
  replicas: 2
```

```

selector:
  matchLabels:
    app: nginx-emptydir
template:
  metadata:
    labels:
      app: nginx-emptydir
  spec:
    containers:
      - name: container-1
        image: nginx:latest
        volumeMounts:
          - name: vol-emptydir # Volume name, which must be the same as the
            # volume name in the volumes field.
            mountPath: /tmp # Path to which an EV is mounted.
        imagePullSecrets:
          - name: default-secret
        volumes:
          - name: vol-emptydir # Volume name, which can be customized.
            emptyDir:
              medium: Memory # EV disk medium: If this parameter is set to
              # Memory, the memory is enabled. If this parameter is left blank, the native
              # default storage medium is used.
              sizeLimit: 1Gi # Volume capacity.
    
```

**Paso 3** Cree una carga de trabajo.

**kubectl apply -f nginx-emptydir.yaml**

----Fin

## EV local

Local Ephemeral Volumes (EVs) se almacenan en los [grupos de almacenamiento](#) de EV. Los EV locales ofrecen un mejor rendimiento que el medio de almacenamiento predeterminado de emptyDir nativo y admiten la expansión horizontal.

### Usando la consola

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** En el panel de navegación de la izquierda, haga clic en **Workloads**. En el panel derecho, haga clic en la ficha **Deployments**.

**Paso 3** Haga clic en **Create Workload** en la esquina superior derecha de la página. En el área **Container Settings**, haga clic en la ficha **Data Storage** y haga clic en **Add Volume > Local Ephemeral Volume (emptyDir)**.

**Paso 4** Montar y utilizar volúmenes de almacenamiento, como se muestra en [Tabla 8-44](#). Para obtener más información sobre otros parámetros, consulte [Workloads](#).

**Tabla 8-44** Montaje de un EV local

Parámetro	Descripción
Capacity	Capacidad del volumen de almacenamiento solicitado.

Parámetro	Descripción
Add Container Path	<p>Configure los parámetros siguientes:</p> <ol style="list-style-type: none"> <li><b>Mount Path:</b> Introduzca una ruta de acceso del contenedor. Por ejemplo, <code>/tmp</code>.                      Este parámetro indica la ruta de contenedor en la que se montará un volumen de datos. No monte el volumen en un directorio del sistema como <code>/</code> o <code>/var/run</code>; esta acción puede causar errores de contenedor. Se recomienda montar el volumen en un directorio vacío. Si el directorio no está vacío, asegúrese de que no hay archivos que afecten al inicio de contenedor. De lo contrario, se reemplazarán los archivos, lo que provocará errores de inicio de contenedor o errores de creación de carga de trabajo.</li> </ol> <p><b>AVISO</b>                      Cuando el contenedor está montado en un directorio de alto riesgo, se recomienda utilizar una cuenta con los permisos mínimos para iniciar el contenedor; de lo contrario, los archivos de alto riesgo en la máquina host podrían estar dañados.</p> <ol style="list-style-type: none"> <li><b>Subpath:</b> Introduzca una ruta secundaria, por ejemplo, <code>tmp</code>.                      Se utiliza una ruta secundaria para montar un volumen local de modo que se utilice el mismo volumen de datos en un solo pod. Si este parámetro se deja en blanco, la ruta raíz se utiliza de forma predeterminada.</li> <li><b>Permission</b> <ul style="list-style-type: none"> <li><b>Read-only:</b> Solo puede leer los datos en los volúmenes montados.</li> <li><b>Read/Write:</b> Puede modificar los volúmenes de datos montados en la ruta de acceso. Los datos recién escritos no se migran si se migra el contenedor, lo que puede causar la pérdida de datos.</li> </ul> </li> </ol> <p>Puede hacer clic en  para agregar varias rutas y subrutas.</p>

**Paso 5** Después de configurar otros parámetros de carga de trabajo, haga clic en **Create**.

----Fin

### Usando kubectl

**Paso 1** Utilice kubectl para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).

**Paso 2** Cree un archivo llamado `nginx-emptydir.yaml` y edítelo.

#### vi `nginx-emptydir.yaml`

Contenido del archivo YAML:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-emptydir
  namespace: default
spec:
  replicas: 2
```

```

selector:
  matchLabels:
    app: nginx-emptydir
template:
  metadata:
    labels:
      app: nginx-emptydir
  spec:
    containers:
      - name: container-1
        image: nginx:latest
        volumeMounts:
          - name: vol-emptydir # Volume name, which must be the same as
the volume name in the volumes field.
            mountPath: /tmp # Path to which an EV is mounted.
        imagePullSecrets:
          - name: default-secret
        volumes:
          - name: vol-emptydir # Volume name, which can be customized.
            emptyDir:
              medium: LocalVolume # If the disk medium of emptyDir is set
to LocalVolume, the local EV is used.
              sizeLimit: 1Gi # Volume capacity.
    
```

**Paso 3** Cree una carga de trabajo.

```
kubectl apply -f nginx-emptydir.yaml
```

----Fin

## Manejo de excepciones de EV locales

Si un usuario desconecta manualmente un disco de ECS o ejecuta manualmente el comando **vgremove**, el grupo de almacenamiento de EV puede funcionar mal. Para resolver este problema, configure el nodo para que no se pueda programar siguiendo el procedimiento descrito en [Configuración de programación de nodos](#) y, a continuación, restablezca el nodo.

## 8.9 hostPath

hostPath se utiliza para montar el directorio de archivos del host donde se encuentra el contenedor en el punto de montaje especificado del contenedor. Si el contenedor necesita tener acceso a **/etc/hosts**, utilice hostPath para asignar **/etc/hosts**.

### AVISO

- Evite usar volúmenes de hostPath tanto como sea posible, ya que son propensos a riesgos de seguridad. Si se deben utilizar volúmenes hostPath, solo se pueden aplicar a archivos o rutas de acceso y montar en modo de solo lectura.
- Después de eliminar el pod en el que se monta un volumen hostPath, se conservan los datos del volumen hostPath.

## Montar un volumen de hostPath en la consola

Puede montar una ruta de acceso en el host a una ruta de acceso de contenedor especificada. Un volumen hostPath se usa generalmente para **almacenar permanentemente los logs de carga de trabajo** o para cargas de trabajo que necesitan **acceder a la estructura de datos interna del motor Docker en el host**.

- Paso 1** Inicie sesión en la consola de CCE.
- Paso 2** Cuando cree una carga de trabajo, haga clic en **Data Storage** en **Container Settings**. Haga clic en **Add Volume** y elija **hostPath** en la lista desplegable.
- Paso 3** Establezca los parámetros para agregar un volumen local, tal como aparece en [Tabla 8-45](#).

**Tabla 8-45** Configuración de parámetros para el montaje de un volumen hostPath

Parámetro	Descripción
Storage Type	Seleccione <b>HostPath</b> .
Host Path	<p>Ruta del host en el que se va a montar el volumen local, por ejemplo, <b>/etc/hosts</b>.</p> <p><b>NOTA</b></p> <p><b>Host Path</b> no se puede establecer en el directorio raíz <b>/</b>. De lo contrario, el montaje falla. Las rutas de montaje pueden ser las siguientes:</p> <ul style="list-style-type: none"> <li>● <b>/opt/xxxx</b> (excepto <b>/opt/cloud</b>)</li> <li>● <b>/mnt/xxxx</b> (excepto <b>/mnt/paas</b>)</li> <li>● <b>/tmp/xxx</b></li> <li>● <b>/var/xxx</b> (excluidos directorios clave como <b>/var/lib</b>, <b>/var/script</b> y <b>/var/paas</b>)</li> <li>● <b>/xxxx</b> (No puede entrar en conflicto con el directorio del sistema, como <b>bin</b>, <b>lib</b>, <b>home</b>, <b>root</b>, <b>boot</b>, <b>dev</b>, <b>etc</b>, <b>lost+found</b>, <b>mnt</b>, <b>proc</b>, <b>sbin</b>, <b>srv</b>, <b>tmp</b>, <b>var</b>, <b>media</b>, <b>opt</b>, <b>selinux</b>, <b>sys</b> y <b>usr</b>.)</li> </ul> <p>No ajuste este parámetro a <b>/home/paas</b>, <b>/var/paas</b>, <b>/var/lib</b>, <b>/var/script</b>, <b>/mnt/paas</b> o <b>/opt/cloud</b>. De lo contrario, la instalación del sistema o del nodo fallará.</p>

Parámetro	Descripción
Add Container Path	<p>Configure los parámetros siguientes:</p> <ol style="list-style-type: none"> <li><b>Mount Path:</b> Introduzca una ruta de acceso del contenedor. Por ejemplo, <code>/tmp</code>. Este parámetro indica la ruta de contenedor en la que se montará un volumen de datos. No monte el volumen en un directorio del sistema como <code>/o /var/run</code>; esta acción puede causar errores de contenedor. Se recomienda montar el volumen en un directorio vacío. Si el directorio no está vacío, asegúrese de que no hay archivos que afecten al inicio de contenedor. De lo contrario, se reemplazarán los archivos, lo que provocará errores de inicio de contenedor o errores de creación de carga de trabajo.</li> </ol> <p><b>AVISO</b>                      Cuando el contenedor está montado en un directorio de alto riesgo, se recomienda utilizar una cuenta con los permisos mínimos para iniciar el contenedor; de lo contrario, los archivos de alto riesgo en la máquina host podrían estar dañados.</p> <ol style="list-style-type: none"> <li><b>Subpath:</b> Introduzca una ruta secundaria, por ejemplo, <code>tmp</code>. Se utiliza una ruta secundaria para montar un volumen local de modo que se utilice el mismo volumen de datos en un solo pod. Si este parámetro se deja en blanco, la ruta raíz se utiliza de forma predeterminada.</li> <li><b>Permission</b> <ul style="list-style-type: none"> <li><b>Read-only:</b> Solo puede leer los datos en los volúmenes montados.</li> <li><b>Read/Write:</b> Puede modificar los volúmenes de datos montados en la ruta de acceso. Los datos recién escritos no se migran si se migra el contenedor, lo que puede causar la pérdida de datos.</li> </ul> </li> </ol> <p>Puede hacer clic en  para agregar varias rutas y subrutas.</p>

**Paso 4** Después de completar la configuración, haga clic en **Create**.

---Fin

## Montar un volumen hostPath usando kubectl

**Paso 1** Utilice kubectl para conectarse al clúster.

**Paso 2** Cree un archivo llamado `nginx-hostpath.yaml` y editelo.

**vi nginx-hostpath.yaml**

El contenido del archivo YAML es el siguiente. Monte el directorio `/data` en el nodo en el directorio `/data` del nodo.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-hostpath
  namespace: default
spec:
```

```
replicas: 2
selector:
  matchLabels:
    app: nginx-hostpath
template:
  metadata:
    labels:
      app: nginx-hostpath
  spec:
    containers:
      - name: container-1
        image: nginx:latest
        volumeMounts:
          - name: vol-hostpath          # Volume name, which must be the same as
the volume name in the volumes field.
            mountPath: /data          # Mount path in the container.
        imagePullSecrets:
          - name: default-secret
        volumes:
          - name: vol-hostpath          # Volume name, which can be customized.
            hostPath:
              path: /data          # Directory location on the host node.
```

**Paso 3** Cree una carga de trabajo.

```
kubectl apply -f nginx-hostpath.yaml
```

----Fin

## 8.10 StorageClass

### Presentación

StorageClass describe la clasificación de tipos de almacenamiento en un clúster y se puede representar como una plantilla de configuración para crear los PV. Al crear un PVC o un PV, debe especificar StorageClass.

Como usuario, solo necesita especificar **storageClassName** al definir un PVC para crear automáticamente un PV y almacenamiento subyacente, lo que reduce significativamente la carga de trabajo de creación y mantenimiento de un PV.

Además de las **clases de almacenamiento predeterminadas** proporcionadas por CCE, también puede personalizar las clases de almacenamiento.

- [Escenarios de aplicaciones de almacenamiento personalizado](#)
- [Clase de almacenamiento personalizado](#)
- [Especificación de un StorageClass predeterminado](#)
- [Especificación de un proyecto de empresa para clases de almacenamiento](#)

### Clases de almacenamiento predeterminadas de CCE

A partir de ahora, CCE proporciona clases de almacenamiento como `csi-disk`, `csi-nas` y `csi-obs` de forma predeterminada. Al definir un PVC, puede utilizar un **storageClassName** para crear automáticamente un PV del tipo correspondiente y crear automáticamente recursos de almacenamiento subyacentes.

Puede ejecutar el siguiente comando `kubectl` para consultar las clases de almacenamiento que admite CCE. Puede utilizar el complemento de CSI proporcionado por CCE para crear una clase de almacenamiento.

```
# kubectl get sc
NAME                PROVISIONER          AGE          # Storage class
csi-disk            everest-csi-provisioner  17d
for EVS disks
csi-nas             everest-csi-provisioner  17d          # Storage class
for SFS 1.0 file systems
csi-obs             everest-csi-provisioner  17d          # Storage class
for OBS buckets
csi-sfsturbo       everest-csi-provisioner  17d          # Storage class
for SFS Turbo file systems
csi-local-topology everest-csi-provisioner  17d          # Local PV
```

Cada clase de almacenamiento contiene los parámetros predeterminados usados para crear dinámicamente un PV. A continuación se muestra un ejemplo de clase de almacenamiento para los discos de EVS:

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: csi-disk
provisioner: everest-csi-provisioner
parameters:
  csi.storage.k8s.io/csi-driver-name: disk.csi.everest.io
  csi.storage.k8s.io/fstype: ext4
  everest.io/disk-volume-type: SAS
  everest.io/passthrough: 'true'
reclaimPolicy: Delete
allowVolumeExpansion: true
volumeBindingMode: Immediate
```

Parámetro	Descripción
provisioner	Especifica el proveedor de recursos de almacenamiento, que es el complemento más antiguo para CCE. Establezca este parámetro en <b>everest-csi-provisioner</b> .
parameters	Especifica los parámetros de almacenamiento, que varían según los tipos de almacenamiento.
reclaimPolicy	Especifica el valor de <b>persistentVolumeReclaimPolicy</b> para crear un PV. El valor puede ser <b>Delete</b> o <b>Retain</b> . Si no se especifica <b>reclaimPolicy</b> al crear un objeto de StorageClass, el valor predeterminado es <b>Delete</b> . <ul style="list-style-type: none"> <li>● <b>Delete</b>: indica que un PV creado dinámicamente se destruirá automáticamente.</li> <li>● <b>Retain</b>: indica que un PV creado dinámicamente no se destruirá automáticamente.</li> </ul>
allowVolumeExpansion	Especifica si el PV de esta clase de almacenamiento admite la expansión de capacidad dinámica. El valor predeterminado es <b>false</b> . La ampliación de la capacidad dinámica se implementa mediante el complemento de almacenamiento subyacente. Esto es solo un interruptor.

Parámetro	Descripción
volumeBindingMode	<p>Especifica el modo de enlace de volumen, es decir, el momento en que se crea dinámicamente un PV. El valor puede ser <b>Immediate</b> o <b>WaitForFirstConsumer</b>.</p> <ul style="list-style-type: none"> <li>● <b>Immediate</b>: la unión de PV y la creación dinámica se completan cuando se crea un PVC.</li> <li>● <b>WaitForFirstConsumer</b>: la unión y la creación de PV se retrasan. Los procesos de la creación y la unión de PV solo se ejecutan cuando el PVC se utiliza en la carga de trabajo.</li> </ul>
mountOptions	<p>Este campo debe ser compatible con el almacenamiento subyacente. Si este campo no se admite pero se especifica, se producirá un error en la creación de PV.</p>

## Escenarios de aplicaciones de almacenamiento personalizado

Cuando se utilizan recursos de almacenamiento en CCE, el método más común es especificar **storageClassName** para definir el tipo de recursos de almacenamiento que se crearán al crear un PVC. La siguiente configuración muestra cómo utilizar un PVC para solicitar un disco de EVS SAS (capacidad alta de E/S) (almacenamiento en bloque).

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-evs-example
  namespace: default
  annotations:
    everest.io/disk-volume-type: SAS
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
  storageClassName: csi-disk
```

Si necesita especificar el tipo de disco de EVS en CCE, utilice el campo **everest.io/disk-volume-type**. SAS indica el tipo del disco de EVS.

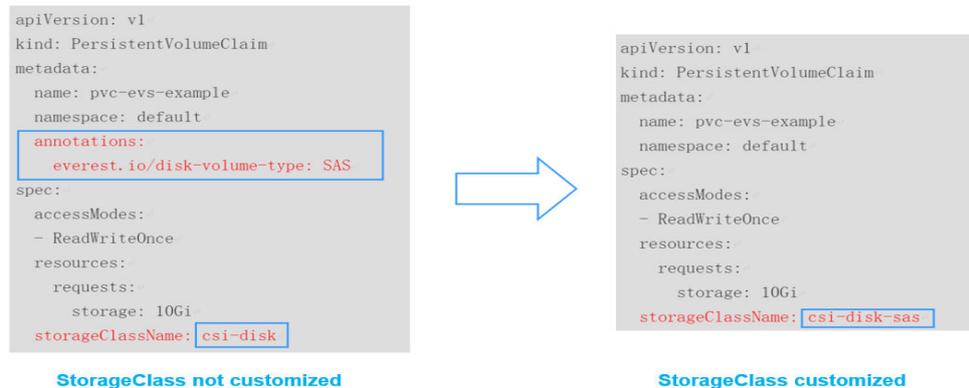
Lo anterior es un método básico de usar StorageClass. En escenarios del mundo real, puede usar StorageClass para realizar otras operaciones.

Escenario de aplicación	Solución	Procedimiento
<p>Cuando se utiliza <b>annotations</b> para especificar la configuración de almacenamiento, la configuración es compleja. Por ejemplo, el campo <b>everest.io/disk-volume-type</b> se utiliza para especificar el tipo del disco de EVS.</p>	<p>Defina las anotaciones de PVC en el campo <b>parameters</b> de StorageClass. Al compilar un archivo YAML, solo tiene que especificar <b>storageClassName</b>.</p> <p>Por ejemplo, puede definir el disco de EVS SAS y el disco EVS SSD como una clase de almacenamiento, respectivamente. Si se define una clase de almacenamiento denominada <b>csi-disk-sas</b>, se utiliza para crear el almacenamiento de SAS.</p>	<p><b>Clase de almacenamiento personalizado</b></p>
<p>Cuando un usuario migra servicios desde un clúster de Kubernetes creado automáticamente u otros servicios de Kubernetes a CCE, la clase de almacenamiento utilizada en el archivo YAML de la aplicación original es diferente de la utilizada en CCE. Como resultado, un gran número de archivos YAML o paquetes de gráficos Helm necesitan ser modificados cuando se utiliza el almacenamiento, que es complejo y propenso a errores.</p>	<p>Cree una clase de almacenamiento con el mismo nombre que en el archivo YAML de la aplicación original en la centralización de CCE. Después de la migración, no es necesario modificar el <b>storageClassName</b> en el archivo YAML de la aplicación.</p> <p>Por ejemplo, la clase de almacenamiento de disco de EVS utilizada antes de la migración es <b>disk-standard</b>. Después de migrar servicios a un clúster de CCE, puede copiar el archivo YAML de la clase de almacenamiento <b>csi-disk</b> en el clúster de CCE, cambiar su nombre a <b>disk-standard</b> y crear otra clase de almacenamiento.</p>	
<p><b>storageClassName</b> debe especificarse en el archivo YAML para usar el almacenamiento. Si no es así, no se puede crear el almacenamiento.</p>	<p>Si establece el StorageClass predeterminado en el clúster, puede crear almacenamiento sin especificar el <b>storageClassName</b> en el archivo YAML.</p>	<p><b>Especificación de un StorageClass predefinido</b></p>

## Clase de almacenamiento personalizado

Esta sección utiliza la clase de almacenamiento personalizado de discos de EVS como ejemplo para describir cómo definir el disco de EVS SAS y el disco de EVS SSD como una clase de almacenamiento, respectivamente. Por ejemplo, si define una clase de almacenamiento denominada **csi-disk-sas** que se utiliza para crear almacenamiento de SAS,

las diferencias se muestran en la siguiente figura. Al compilar un archivo YAML, solo tiene que especificar **storageClassName**.



- Puede personalizar una clase de almacenamiento de capacidad alta de E/S en un archivo YAML. Por ejemplo, el nombre **csi-disk-sas** indica que el tipo de disco es SAS (capacidad alta de E/S).

```

    apiVersion: storage.k8s.io/v1
    kind: StorageClass
    metadata:
      name: csi-disk-sas # Name of the high I/O storage
                        class, which can be customized.
    parameters:
      csi.storage.k8s.io/csi-driver-name: disk.csi.everest.io
      csi.storage.k8s.io/fstype: ext4
      everest.io/disk-volume-type: SAS # High I/O EVS disk type,
      everest.io/passthrough: "true" # which cannot be customized.
    provisioner: everest-csi-provisioner
    reclaimPolicy: Delete
    volumeBindingMode: Immediate
    allowVolumeExpansion: true # true indicates that capacity
                               expansion is allowed.
  
```

- Para una clase de almacenamiento de capacidad ultraalta de E/S, puede establecer el nombre de la clase en **csi-disk-ssd** para crear un disco de EVS SSD (capacidad ultraalta de E/S).

```

    apiVersion: storage.k8s.io/v1
    kind: StorageClass
    metadata:
      name: csi-disk-ssd # Name of the ultra-high I/O
                        storage class, which can be customized.
    parameters:
      csi.storage.k8s.io/csi-driver-name: disk.csi.everest.io
      csi.storage.k8s.io/fstype: ext4
      everest.io/disk-volume-type: SSD # Ultra-high I/O EVS disk type,
      everest.io/passthrough: "true" # which cannot be customized.
    provisioner: everest-csi-provisioner
    reclaimPolicy: Delete
    volumeBindingMode: Immediate
    allowVolumeExpansion: true
  
```

**reclaimPolicy:** indica las políticas de recuperación del almacenamiento en la nube subyacente. El valor puede ser **Delete** o **Retain**.

- **Delete:** Cuando se elimina un PVC, se eliminan tanto el PV como el disco de EVS.
- **Retain:** Cuando se elimina un PVC, el PV y los recursos de almacenamiento subyacentes no se eliminan. En su lugar, debe eliminar manualmente estos recursos. Después de eso, el PV está en el estado **Released** y no puede estar ligado al PVC de nuevo.

Si se requiere una alta seguridad de los datos, se recomienda seleccionar **Retain** para evitar que los datos se eliminen por error.

Una vez completada la definición, ejecute los comandos **kubectl create** para crear recursos de almacenamiento.

```
# kubectl create -f sas.yaml
storageclass.storage.k8s.io/csi-disk-sas created
# kubectl create -f ssd.yaml
storageclass.storage.k8s.io/csi-disk-ssd created
```

Consulte **StorageClass** de nuevo. El resultado del comando es el siguiente:

```
# kubectl get sc
NAME                PROVISIONER          AGE
csi-disk            everest-csi-provisioner  17d
csi-disk-sas       everest-csi-provisioner  2m28s
csi-disk-ssd       everest-csi-provisioner  16s
csi-disk-topology  everest-csi-provisioner  17d
csi-nas            everest-csi-provisioner  17d
csi-obs            everest-csi-provisioner  17d
csi-sfsturbo       everest-csi-provisioner  17d
```

## Especificación de un StorageClass predeterminado

Puede especificar una clase de almacenamiento como clase predeterminada. De esta manera, si no especifica **storageClassName** al crear un PVC, el PVC se crea utilizando la clase de almacenamiento predeterminada.

Por ejemplo, para especificar **csi-disk-ssd** como la clase de almacenamiento predeterminada, edite el archivo YAML de la siguiente manera:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-disk-ssd
  annotations:
    storageclass.kubernetes.io/is-default-class: "true" # Specifies the default
storage class in a cluster. A cluster can have only one default storage class.
parameters:
  csi.storage.k8s.io/csi-driver-name: disk.csi.everest.io
  csi.storage.k8s.io/fstype: ext4
  everest.io/disk-volume-type: SSD
  everest.io/passthrough: "true"
provisioner: everest-csi-provisioner
reclaimPolicy: Delete
volumeBindingMode: Immediate
allowVolumeExpansion: true
```

Elimine el disco **csi-disk-ssd** creado, ejecute el comando **kubectl create** para crear un disco **csi-disk-ssd** de nuevo y, a continuación, consulte la clase de almacenamiento. La siguiente información aparecerá en la pantalla.

```
# kubectl delete sc csi-disk-ssd
storageclass.storage.k8s.io "csi-disk-ssd" deleted
# kubectl create -f ssd.yaml
storageclass.storage.k8s.io/csi-disk-ssd created
# kubectl get sc
NAME                PROVISIONER          AGE
csi-disk            everest-csi-provisioner  17d
csi-disk-sas       everest-csi-provisioner  114m
csi-disk-ssd (default)  everest-csi-provisioner  9s
csi-disk-topology  everest-csi-provisioner  17d
csi-nas            everest-csi-provisioner  17d
csi-obs            everest-csi-provisioner  17d
csi-sfsturbo       everest-csi-provisioner  17d
```

## Especificación de un proyecto de empresa para clases de almacenamiento

CCE permite especificar un proyecto de empresa al crear discos de EVS y PVC de OBS. Los recursos de almacenamiento creados (discos de EVS y OBS) pertenecen al proyecto de empresa especificado. **El proyecto de empresa puede ser el proyecto de empresa al que pertenece el cluster o el proyecto de empresa por defecto.**

Si no especifica ningún proyecto de empresa, el proyecto de empresa de StorageClass se utiliza de forma predeterminada. Los recursos de almacenamiento creados mediante las clases de almacenamiento csi-disk y csi-obs de CCE pertenecen al proyecto de empresa predeterminado.

Si desea que los recursos de almacenamiento creados a partir de las clases de almacenamiento estén en el mismo proyecto de empresa que el clúster, puede personalizar una clase de almacenamiento y especificar el ID del proyecto de empresa, como se muestra a continuación.

### NOTA

Para utilizar esta función, el complemento everest debe actualizarse a 1.2.33 o posterior.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: csi-disk-epid      #Customize a storage class name.
provisioner: everest-csi-provisioner
parameters:
  csi.storage.k8s.io/csi-driver-name: disk.csi.everest.io
  csi.storage.k8s.io/fstype: ext4
  everest.io/disk-volume-type: SAS
  everest.io/enterprise-project-id: 86bfc701-9d9e-4871-a318-6385aa368183
#Specify the enterprise project ID.
  everest.io/passthrough: 'true'
reclaimPolicy: Delete
allowVolumeExpansion: true
volumeBindingMode: Immediate
```

## Verificación

- Utilice **csi-disk-sas** para crear un PVC.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: sas-disk
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
  storageClassName: csi-disk-sas
```

Cree una clase de almacenamiento y vea sus detalles. Como se muestra a continuación, el objeto se puede crear y el valor de **STORAGECLASS** es **csi-disk-sas**.

```
# kubectl create -f sas-disk.yaml
persistentvolumeclaim/sas-disk created
# kubectl get pvc
NAME          STATUS      VOLUME                                     CAPACITY
ACCESS MODES  STORAGECLASS  AGE
sas-disk     Bound       pvc-6e2f37f9-7346-4419-82f7-b42e79f7964c  10Gi
RWO          csi-disk-sas  24s
# kubectl get pv
NAME          CAPACITY  ACCESS MODES  RECLAIM
POLICY  STATUS      CLAIM          STORAGECLASS  REASON  AGE
pvc-6e2f37f9-7346-4419-82f7-b42e79f7964c  10Gi      RWO
```

Delete	Bound	default/sas-disk	csi-disk-
sas	30s		

Vea los detalles de PVC en la consola de CCE. En la página de detalles de PV, puede ver que el tipo de disco es con capacidad alta de E/S.

PV Details		Storage Details		Event	
PV Name	3bd17	Creation Method	Unknown		
PV Status	Bound	Storage Class	EV5 disk		
Access Mode	ReadWriteOnce	Sub-class	High I/O		
PV Reclaim Policy	Delete	Capacity	10 GB		
PV Created	Apr 26, 2021 16:26:57 GMT+08:00	Volume ID	3c956		
PV ID	18de				

- Si no se especifica **storageClassName**, se utiliza la configuración predeterminada, como se muestra a continuación.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: ssd-disk
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
```

Cree y vea el recurso de almacenamiento. Puede ver que la clase de almacenamiento de PVC `ssd-disk` es `csi-disk-ssd`, lo que indica que `csi-disk-ssd` se usa de forma predeterminada.

```
# kubectl create -f ssd-disk.yaml
persistentvolumeclaim/ssd-disk created
# kubectl get pvc
NAME          STATUS    VOLUME                                     CAPACITY
ACCESS MODES STORAGECLASS  AGE
sas-disk     Bound     pvc-6e2f37f9-7346-4419-82f7-b42e79f7964c  10Gi
RWO          csi-disk-sas  16m
ssd-disk     Bound     pvc-4d2b059c-0d6c-44af-9994-f74d01c78731  10Gi
RWO          csi-disk-ssd  10s
# kubectl get pv
NAME          CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS    CLAIM                                     STORAGECLASS  REASON  AGE
pvc-4d2b059c-0d6c-44af-9994-f74d01c78731  10Gi      RWO           Delete          Bound     default/ssd-disk                         csi-disk-ssd  15s
pvc-6e2f37f9-7346-4419-82f7-b42e79f7964c  10Gi      RWO           Delete          Bound     default/sas-disk                         csi-disk-sas  17m
```

Vea los detalles de PVC en la consola de CCE. En la página de detalles del PV, puede ver que el tipo de disco es con capacidad ultraalta de E/S.

PV Details		Storage Details		Event	
PV Name	pvc-39e488d1-07af-4590-8568-3bd9f9e3bd17	Creation Method	Unknown		
PV Status	Bound	Storage Class	EV5 disk		
Access Mode	ReadWriteOnce	Sub-class	Ultra-High I/O		
PV Reclaim Policy	Delete	Capacity	10 GB		
PV Created	Apr 26, 2021 16:26:57 GMT+08:00	Volume ID	fae10df4-dbf4-476f-8f98-734dddf6c956		
PV ID	55c2d57-5f1b-4011-8c05-d9eed34c980e				

## 8.11 Grupos de almacenamiento

CCE le permite usar LVM para combinar volúmenes de datos en nodos en un grupo de almacenamiento (VolumeGroup) y crear LV para que contenedores los monte. Un grupo de almacenamiento admite los siguientes modos de escritura:

- **Linear:** Un volumen lógico lineal integra uno o más volúmenes físicos. Los datos se escriben en el siguiente volumen físico cuando se agota el anterior.
- **Striped:** Un volumen lógico rayado separa los datos en bloques del mismo tamaño y los almacena en múltiples volúmenes físicos en secuencia, lo que permite que los datos se lean y escriban simultáneamente. Seleccione esta opción solo cuando haya varios volúmenes.

## Restricciones

- El primer disco de datos (usado por contenedor en tiempo de ejecución y el componente de kubelet) en un nodo no se puede importar como un grupo de almacenamiento.
- Según el tipo de volumen, las restricciones en las versiones de clúster y complementos del grupo de almacenamiento que se van a importar son las siguientes:
  - Los PV locales solo se admiten cuando la versión del clúster es v1.21.2-r0 o posterior y la versión adicional más reciente es 2.1.23 o posterior. Se recomienda 2.1.23 o posterior.
  - Los EV locales solo se admiten cuando la versión del clúster es v1.21.2-r0 o posterior y la versión adicional es 1.2.29 o posterior.
- Los grupos de almacenamiento en modo seccionado no admiten el escalamiento horizontal. Después del escalado horizontal, se puede generar espacio fragmentado y no se puede usar el grupo de almacenamiento.
- Los grupos de almacenamiento no se pueden escalar ni eliminar.
- Si se eliminan los discos de un grupo de almacenamiento en un nodo, el grupo de almacenamiento funcionará mal.

## Importación de un grupo de almacenamiento

### Importado durante la creación del nodo

Al crear un nodo, puede agregar un disco de datos al nodo en la configuración de almacenamiento e importar el disco de datos al grupo de almacenamiento como un PV o EV. Para obtener más información, véase [Creación de un nodo](#).

**Storage Settings** Configure storage resources for containers and applications on the node.

System Disk: High I/O, 50 GB, Expand

Data Disk: High I/O, 100 GB, Expand

Used by the container runtime and kubelet. Do not uninstall this disk. Otherwise, the node will become unavailable. [How do I set data disk size?](#) [How do I allocate data disk space?](#)

High I/O, 100 GB, Hide

For a common data disk, you can choose not to perform any operation (by default) or attach it in a specified mode.

**Mount Settings**

Default  Mount Disk  Use as PV  Use as ephemeral volume

**Data Disk Encryption**

Encryption

**Add Data Disk** Available for creation: 3

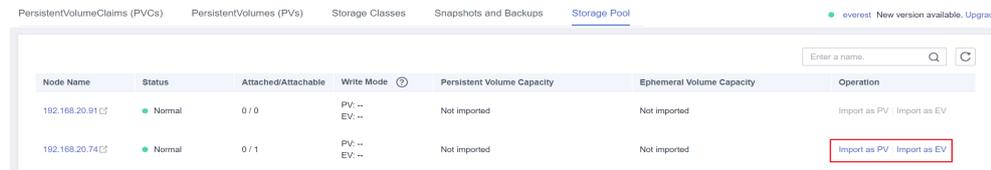
**PV Write**

Linear  Striped

### Importado manualmente

Si no se importa ningún PV o EV durante la creación del nodo, o la capacidad del volumen de almacenamiento actual es insuficiente, puede importar manualmente un grupo de almacenamiento.

- Paso 1** Vaya a la consola de ECS y agregue un disco de SCSI al nodo. Para más detalles, consulte [Adición de un disco](#).
- Paso 2** Inicie sesión en la consola de CCE y acceda a la consola del clúster.
- Paso 3** En el panel de navegación, elija **Storage** y cambie a la ficha **Storage Pool**.
- Paso 4** Vea el nodo al que se ha agregado el disco y seleccione **Import as PV** o **Import as EV**. Puede seleccionar un modo de escritura durante la importación.



---Fin

## 8.12 Instantáneas y copias de respaldo

CCE trabaja con EVS para admitir instantáneas. Una instantánea es una copia o una imagen completa de los datos del disco de EVS en un punto determinado de tiempo, que se puede utilizar para la recuperación ante desastres de datos.

Puede crear instantáneas para guardar rápidamente los datos del disco en un momento determinado. Además, puede utilizar instantáneas para crear discos de modo que los discos creados contengan los datos de instantáneas al principio.

### Precauciones

- La función de instantánea está disponible **solo para los clústeres de v1.15 o posterior** y requiere el complemento everest basado en CSI.
- El subtipo (E/S común, E/S alta o E/S ultraalta), modo de disco (SCSI o VBD), encriptación de datos, estado de uso compartido, y la capacidad de un disco de EVS creado a partir de una instantánea debe ser la misma que la del disco asociado a la instantánea. Estos atributos no se pueden modificar después de ser consultados o establecidos.
- Las instantáneas solo se pueden crear para discos de CSI disponibles o en uso. Durante la prueba gratuita, puede crear hasta 7 instantáneas por disco.
- Los datos instantáneos de los discos cifrados se almacenan cifrados, y los de los discos no cifrados se almacenan no cifrados.

### Escenario

La función de instantánea ayuda a abordar las siguientes necesidades:

- **Copia de seguridad de datos rutinaria**  
 Puede crear instantáneas para los discos de EVS con regularidad y usar instantáneas para recuperar sus datos en caso de que se produzca una pérdida de datos o una incoherencia de datos debido a un mal funcionamiento, virus o ataques.
- **Restauración rápida de datos**  
 Puede crear una instantánea o varias instantáneas antes de un cambio de SO, una actualización de software de aplicación o una migración de datos de servicio. Si se

produce una excepción durante la actualización o migración, los datos de servicio se pueden restaurar rápidamente en el momento en que se creó la instantánea.

Por ejemplo, se produjo un fallo en el disco A del sistema de ECS A, y por lo tanto ECS A no se puede iniciar. Debido a que el disco A del sistema ya está defectuoso, los datos del disco A del sistema no se pueden restaurar revirtiendo instantáneas. En este caso, puede utilizar una instantánea existente del disco A del sistema para crear el disco B de EVS y conectarlo a ECS B que se esté ejecutando correctamente. Entonces, el ECS B puede leer datos del disco A del sistema usando el disco B del EVS.

#### **NOTA**

La capacidad de instantáneas proporcionada por CCE es la misma que la función de instantáneas de CSI proporcionada por la comunidad de Kubernetes. Los discos de EVS solo se pueden crear basándose en instantáneas, y las instantáneas no se pueden volver a los discos de EVS de origen.

- **Rápida implementación de múltiples servicios**

Puede utilizar una instantánea para crear varios discos de EVS que contengan los mismos datos iniciales, y estos discos se pueden utilizar como recursos de datos para varios servicios, por ejemplo, minería de datos, consulta de informes y desarrollo y pruebas. Este método protege los datos iniciales y crea discos rápidamente, cumpliendo los requisitos de datos de servicio diversificados.

## Creación de una instantánea

### Usar la consola de CCE

**Paso 1** Inicie sesión en la consola de CCE.

**Paso 2** Haga clic en el nombre del clúster y vaya a la consola del clúster. Elija **Storage** en el panel de navegación y haga clic en la ficha **Snapshots and Backups**.

**Paso 3** Haga clic en **Create Snapshot** en la esquina superior derecha. En el cuadro de diálogo que se muestra, defina los parámetros relacionados.

- **Snapshot Name:** Ingrese un nombre de instantánea.
- **Storage:** Seleccione un PVC. Las instantáneas solo se pueden usar para crear PVC de EVS.

**Paso 4** Haga clic en **Create**.

----Fin

### Usar YAML

```
kind: VolumeSnapshot
apiVersion: snapshot.storage.k8s.io/v1beta1
metadata:
  finalizers:
    - snapshot.storage.kubernetes.io/volumesnapshot-as-source-protection
    - snapshot.storage.kubernetes.io/volumesnapshot-bound-protection
  name: cce-disksnap-test
  namespace: default
spec:
  source:
    persistentVolumeClaimName: pvc-eva-test # PVC name. Only an EVS PVC can
    be created.
  volumeSnapshotClassName: csi-disk-snapclass
```

## Uso de una instantánea para crear un PVC

El tipo de disco, la configuración de encriptación, y el modo de disco del PVC de EVS creado son consistentes con los del disco de EVS de origen de la instantánea.

### Usar la consola de CCE

- Paso 1** Inicie sesión en la consola de CCE.
- Paso 2** Haga clic en el nombre del clúster y vaya a la consola del clúster. Elija **Storage** en el panel de navegación y haga clic en la ficha **Snapshots and Backups**.
- Paso 3** Busque la instantánea que desea utilizar para crear un PVC, haga clic en **Create PVC** y especifique el nombre del PVC en el cuadro de diálogo que se muestra.
- Paso 4** Haga clic en **Create**.

----Fin

### Usar YAML

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-test
  namespace: default
  annotations:
    everest.io/disk-volume-type: SSD      # EVS disk type, which must be the same
as that of the source EVS disk of the snapshot.
  labels:
    failure-domain.beta.kubernetes.io/region: ap-southeast-1
    failure-domain.beta.kubernetes.io/zone: ap-southeast-1a
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
  storageClassName: csi-disk
  dataSource:
    name: cce-disksnap-test              # Snapshot name.
    kind: VolumeSnapshot
    apiGroup: snapshot.storage.k8s.io
```

# 9 Monitoreo y alarma

## 9.1 Resumen de monitoreo

CCE trabaja con AOM para monitorear de forma integral los clústeres. Cuando se crea un nodo, el ICAgent (el DaemonSet llamado **icagent** en el espacio de nombres del kube-system del clúster) de AOM está instalado de forma predeterminada. ICAgent recopila datos de supervisión de los recursos y cargas de trabajo subyacentes que se ejecutan en el clúster. También recopila datos de monitoreo de métricas personalizadas de la carga de trabajo.

- Métricas de recursos

El monitoreo básico de recursos incluye monitoreo a CPU, a memoria y a disco. Para obtener más información, véase [Métricas de recursos](#). Puede ver estas métricas de clústeres, nodos y cargas de trabajo en la consola de CCE o de AOM.

- Métricas personalizadas

El ICAgent recopila métricas personalizadas de aplicaciones y las carga en AOM. Para obtener más información, véase [Supervisión de métricas personalizadas en AOM](#).

- Monitorización de NPD

node-problem-detector (npd para abreviar) es un complemento que monitorea e informa sobre el estado de un nodo. Se puede conectar a una plataforma de monitoreo de terceros. Es un demonio que se ejecuta en cada nodo. Recopila problemas de nodos de diferentes demonios y los informa al servidor de API. El complemento npd puede ejecutarse como un demonio o un DaemonSet.

CCE mejora npd en la versión 1.16.0, que ahora admite comprobaciones de recursos de nodo, componentes y eventos, así como aislamiento de fallas. Para obtener más información, véase [npd](#).

Además, puede instalar el complemento de Prometheus en un clúster y usar Prometheus para recopilar y mostrar datos de supervisión. Para obtener más información, véase [Monitoreo de métricas personalizadas con prometheus](#).

### Métricas de recursos

En la consola de CCE, puede ver las siguientes métricas.

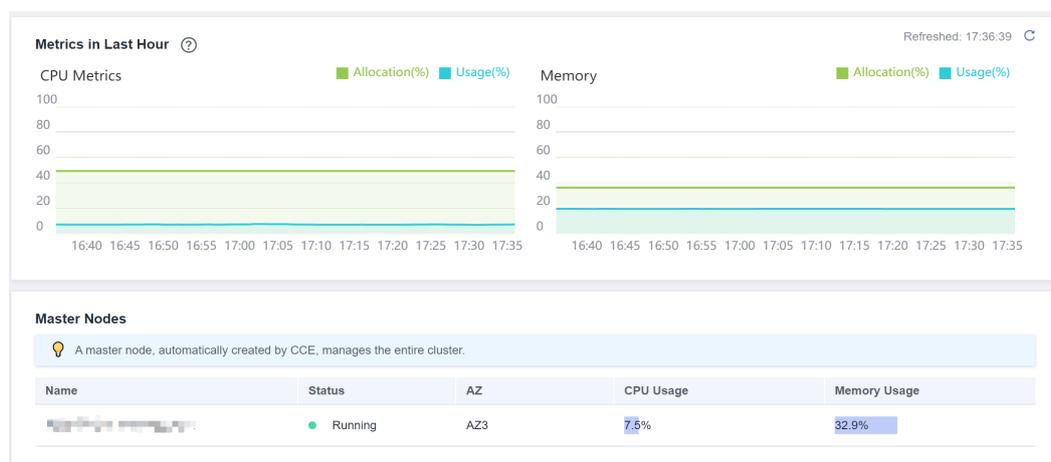
**Tabla 9-1** Métricas de recursos

Métrica	Descripción
CPU Allocation Rate	Indica el porcentaje de CPUs asignadas a cargas de trabajo.
Memory Allocation Rate	Indica el porcentaje de memoria asignada a las cargas de trabajo.
CPU Usage	Indica la utilización del CPU.
Memory Usage	Indica el uso de la memoria.
Disk Usage	Indica el uso del disco.
Down	Indica la velocidad a la que se descargan los datos en un nodo. La unidad es KB/s.
Up	Indica la velocidad a la que se cargan los datos desde un nodo. La unidad es KB/s.
Disk Read Rate	Indica el volumen de datos leído de un disco por segundo. La unidad es KB/s.
Disk Write Rate	Indica el volumen de datos escrito en un disco por segundo. La unidad es KB/s.

En la consola de AOM, puede ver las métricas del host y las métricas de contenedor. Para obtener más información, consulte [Descripción de métrica](#).

## Consulta de datos de supervisión de clústeres

Haga clic en el nombre del clúster y acceda a la consola del clúster. En el panel de navegación, elija **Cluster Information**. En el panel derecho, puede ver el uso de CPU y memoria de todos los nodos (excepto los nodos maestros) del clúster en la última hora.



### Explicación de las métricas de monitorización:

- Tasa de asignación de CPU = Suma de cuotas de CPU solicitadas por los pods en el clúster/Suma de cuotas de CPU que se pueden asignar a todos los nodos (excepto los nodos maestros) en el clúster

- Tasa de asignación de memoria = Suma de cuotas de memoria solicitadas por los pods en el clúster/Suma de cuotas de memoria que se pueden asignar a todos los nodos (excepto los nodos maestros) en el clúster
- Uso de CPU: Uso promedio de CPU de todos los nodos (excepto los nodos maestros) en un clúster
- Uso de memoria: Uso medio de memoria de todos los nodos (excepto los nodos maestros) en un clúster

**NOTA**

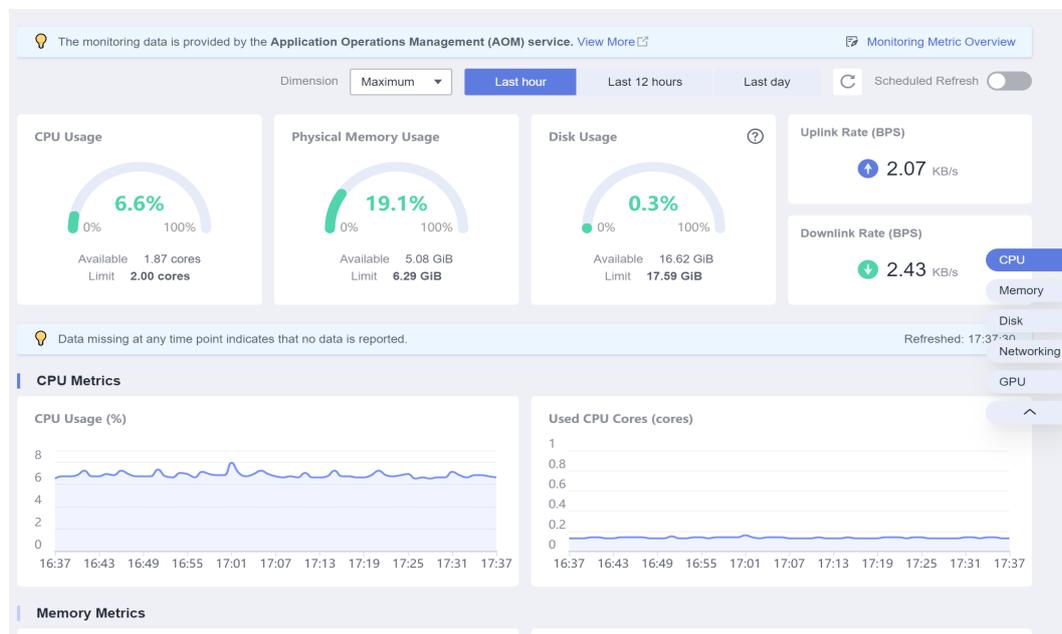
Recursos de nodo asignables (CPU o memoria) = Importe total - Importe reservado - Umbrales de desalojo. Para obtener más información, véase [Descripción de los recursos de nodos reservados](#).

CCE proporciona el estado, la zona de disponibilidad (AZ), el uso de CPU y el uso de memoria de los nodos maestros.

### Consulta de datos de supervisión de nodos de trabajo

Además de ver los datos de supervisión de todos los nodos, también puede ver los datos de supervisión de un solo nodo. Haga clic en el nombre del clúster y acceda a la consola del clúster. Elija **Nodes** en el panel de navegación y haga clic en **Monitor** en la columna **Operation** del nodo de destino.

Los datos de monitoreo provienen de AOM. Puede ver los datos de supervisión de un nodo, incluidos la CPU, la memoria, el disco, la red y la GPU.



### Consulta de datos de supervisión de carga de trabajo

Puede ver los datos de supervisión de una carga de trabajo en la página de ficha **Monitoring** de la página de detalles de la carga de trabajo. Haga clic en el nombre del clúster y acceda a la consola del clúster. Elija **Workloads** en el panel de navegación y haga clic en **Monitor** en la columna **Operation** de la carga de trabajo de destino.

Los datos de monitoreo provienen de AOM. Puede ver los datos de supervisión de una carga de trabajo, incluida la CPU, la memoria, la red y la GPU, en la consola de la unidad de procesamiento.

**Explicación de las métricas de monitorización:**

- Carga de trabajo Uso de CPU = Uso máximo de CPU en cada pod de la carga de trabajo
- Uso de memoria de carga de trabajo = Uso máximo de memoria en cada pod de la carga de trabajo

También puede hacer clic en **View More** para ir a la consola de AOM y ver los datos de supervisión de la carga de trabajo.

## Consulta de datos de supervisión de pod

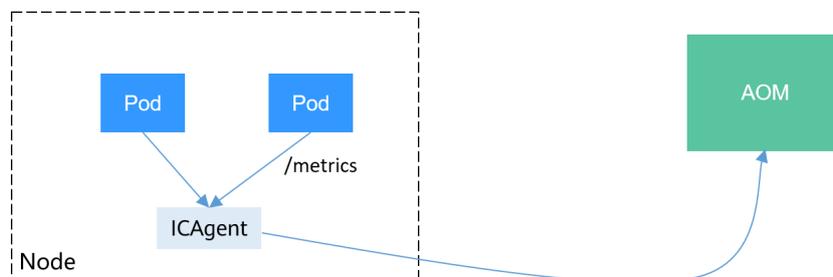
Puede ver los datos de supervisión de un pod en la página de ficha **Pods** de la página de detalles de la carga de trabajo.

**Explicación de las métricas de monitorización:**

- Uso de CPU de pod = Los núcleos de CPU usados/La suma de todos los límites de CPU de los pods (Si no se especifica, se usan todos los núcleos de CPU de nodo.)
- Uso de memoria de pod = La memoria física usada/La suma de todos los límites de memoria de pods (Si no se especifica, se usa toda la memoria de nodo.)

## 9.2 Supervisión de métricas personalizadas en AOM

CCE le permite cargar métricas personalizadas a AOM. El ICAgent en un nodo invoca periódicamente a la API de supervisión de métricas configurada en una carga de trabajo para leer datos de supervisión y luego carga los datos en AOM.



La API métrica personalizada de una carga de trabajo se puede configurar cuando se crea la carga de trabajo. Esta sección utiliza una aplicación de Nginx como ejemplo para describir cómo informar métricas personalizadas a AOM.

### Notas y restricciones

- El ICAgent es compatible con las especificaciones de datos de monitorización de **Prometheus**. Las métricas personalizadas proporcionadas por los pods pueden ser recopiladas por el ICAgent solo cuando cumplen con las especificaciones de datos de monitoreo de Prometheus. Para obtener más información, véase **Recopilación de datos de monitorización de Prometheus**.
- El ICAgent solo admite métricas de **Gauge**.
- El intervalo para que ICAgent invoque a la API de métrica personalizada es de 1 minuto, que no se puede cambiar.

## Recopilación de datos de monitorización de Prometheus

Prometheus invoca periódicamente a la API de monitorización métrica (`/metrics` por defecto) de una aplicación para obtener datos de monitorización. La aplicación debe proporcionar la API de monitorización métrica para que Prometheus invoque, y los datos de monitorización deben cumplir con las siguientes especificaciones de Prometheus:

```
# TYPE nginx_connections_active gauge
nginx_connections_active 2
# TYPE nginx_connections_reading gauge
nginx_connections_reading 0
```

Prometheus ofrece clientes en varios idiomas. Para obtener más información sobre los clientes, consulte [Prometheus CLIENT LIBRARIES](#). Para obtener más información sobre cómo desarrollar un exportador, consulte [WRITING EXPORTERS](#). La comunidad de Prometheus ofrece varios exportadores externos que pueden ser utilizados directamente. Para obtener más información, consulte [EXPORTERS AND INTEGRATIONS](#).

## Preparación de una aplicación

Esta sección utiliza Nginx como ejemplo para describir cómo recopilar datos de monitoreo. Las aplicaciones autodesarrolladas necesitan proporcionar API de monitoreo de métricas para que Prometheus pueda invocar. Para obtener más información, véase [Recopilación de datos de monitorización de Prometheus](#).

Nginx tiene un módulo llamado `ngx_http_stub_status_module` que proporciona funciones básicas de monitorización. Puede configurar el archivo `nginx.conf` para proporcionar una API para que los sistemas externos accedan a los datos de supervisión de Nginx. Como se muestra en la siguiente figura, después de agregar la configuración del servidor a `http`, Nginx puede proporcionar una API para que los sistemas externos accedan a los datos de monitorización de Nginx.

```
user nginx;
worker_processes auto;

error_log /var/log/nginx/error.log warn;
pid /var/run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include /etc/nginx/mime.types;
    default_type application/octet-stream;
    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" "$http_x_forwarded_for"';

    access_log /var/log/nginx/access.log main;
    sendfile on;
    #tcp_nopush on;
    keepalive_timeout 65;
    #gzip on;
    include /etc/nginx/conf.d/*.conf;

    server {
        listen 8080;
        server_name localhost;
        location /stub_status {
            stub_status on;
            access_log off;
        }
    }
}
```

```
}
}
```

Guarde la configuración anterior en el archivo **nginx.conf** y utilice la configuración para crear una nueva imagen. El archivo Dockerfile es el siguiente:

```
FROM nginx:1.21.5-alpine
ADD nginx.conf /etc/nginx/nginx.conf
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

Utilice el archivo Dockerfile anterior para crear una imagen y subirla a SWR. El nombre de la imagen es **nginx:exporter**. Para obtener más información sobre cómo cargar una imagen, consulte [Carga de una imagen a través de un cliente de motor de contenedores](#).

**docker build -t nginx:exporter .**

**docker tag nginx:exporter swr.ap-southeast-1.myhuaweicloud.com/dev-container/nginx:exporter**

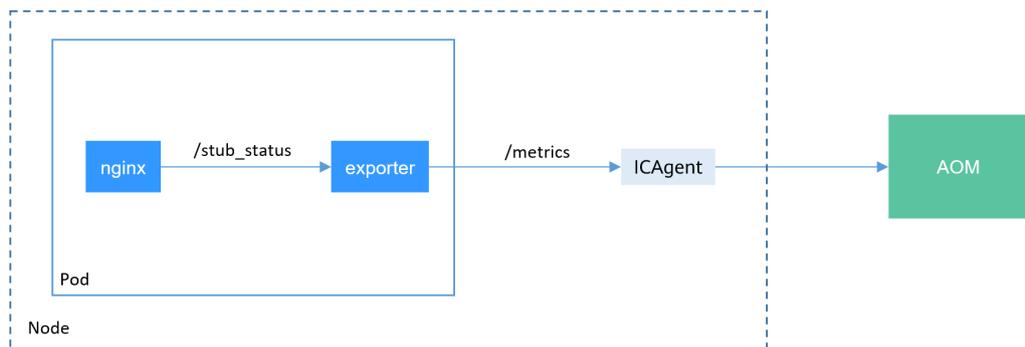
**docker push swr.ap-southeast-1.myhuaweicloud.com/dev-container/nginx:exporter**

Después de ejecutar un contenedor con **nginx:exporter** de imagen, puede obtener datos de monitorización de Nginx llamando a `http://<ip_address> 8080/stub_status`. `<ip_address>` indica la dirección IP del contenedor. Los datos de seguimiento son los siguientes:

```
# curl http://127.0.0.1:8080/stub_status
Active connections: 3
server accepts handled requests
 146269 146269 212
Reading: 0 Writing: 1 Waiting: 2
```

## Despliegue de una aplicación

El formato de datos de los datos de seguimiento proporcionados por **nginx:exporter** no cumple con los requisitos de Prometheus. Es necesario convertir el formato de datos al formato requerido por Prometheus. Para convertir el formato de las métricas de Nginx, utilice **nginx-prometheus-exporter** como se muestra en la siguiente figura.



Despliegue **nginx:exporter** y **nginx-prometheus-exporter** en el mismo pod.

```
kind: Deployment
apiVersion: apps/v1
metadata:
  name: nginx-exporter
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
```

```
    app: nginx-exporter
  template:
    metadata:
      labels:
        app: nginx-exporter
      annotations:
        metrics.alpha.kubernetes.io/custom-endpoints:
' [{"api": "prometheus", "path": "/metrics", "port": "9113", "names": ""}] '
    spec:
      containers:
        - name: container-0
          image: 'nginx:exporter' # Replace it with the address of the image you
uploaded to SWR.
          resources:
            limits:
              cpu: 250m
              memory: 512Mi
            requests:
              cpu: 250m
              memory: 512Mi
        - name: container-1
          image: 'nginx/nginx-prometheus-exporter:0.9.0'
          command:
            - nginx-prometheus-exporter
          args:
            - '-nginx.scrape-uri=http://127.0.0.1:8080/stub_status'
          imagePullSecrets:
            - name: default-secret
```

#### NOTA

La imagen `nginx/nginx-prometheus-exporter:0.9.0` debe extraerse de la red pública. Por lo tanto, cada nodo del clúster debe tener una dirección IP pública.

`nginx-prometheus-exporter` requiere un comando de inicio. Se utiliza **`nginx-prometheus-exporter -nginx.scrape-uri=http://127.0.0.1:8080/stub_status`** para obtener datos de monitorización de Nginx.

Además, debe agregar una anotación **`metrics.alpha.kubernetes.io/custom-endpoints: [{"api": "prometheus", "path": "/metrics", "port": "9113", "names": ""}]`** al pod.

## Verificación

Después de desplegar una aplicación, puede acceder a Nginx para construir algunos datos de acceso y comprobar si los datos de supervisión correspondientes se pueden obtener en AOM.

```
$ kubectl get pod
NAME                                READY   STATUS    RESTARTS   AGE
nginx-exporter-78859765db-6j8sw     2/2    Running   0           4m
$ kubectl exec -it nginx-exporter-78859765db-6j8sw -- /bin/sh
Defaulting container name to container-0.
Use 'kubectl describe pod/nginx-exporter-78859765db-6j8sw -n default' to see all
of the containers in this pod.
/ # curl http://localhost
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
```

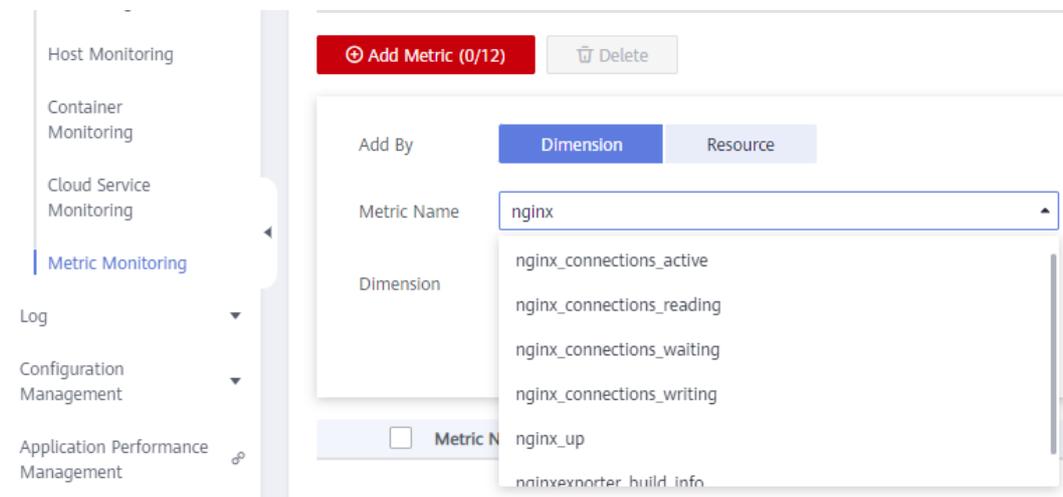
```
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
/ #
```

Puede ver que se ha accedido a Nginx una vez.

Inicie sesión en AOM. En el panel de navegación, elija **Monitoring > Metric Monitoring**. Puede ver las métricas relacionadas con Nginx, por ejemplo, **nginx\_connections\_active**.



## 9.3 Monitoreo de métricas personalizadas con prometheus

Puede utilizar AOM ICAgent para obtener datos métricos personalizados de cargas de trabajo como se describe en [Supervisión de métricas personalizadas en AOM](#). También puede instalar el complemento prometheus en un clúster y usar Prometheus como plataforma de monitoreo.

### Restricciones

Para usar prometheus para monitorear las métricas personalizadas, la aplicación debe proporcionar una API de monitorización de métricas. Para obtener más información, véase [Recopilación de datos de monitorización de Prometheus](#).

### Recopilación de datos de monitorización de Prometheus

Prometheus invoca periódicamente a la API de monitorización métrica (/metrics por defecto) de una aplicación para obtener datos de monitorización. La aplicación debe proporcionar la API de monitorización métrica para que Prometheus invoque, y los datos de monitorización deben cumplir con las siguientes especificaciones de Prometheus:

```
# TYPE nginx_connections_active gauge
nginx_connections_active 2
# TYPE nginx_connections_reading gauge
nginx_connections_reading 0
```

Prometheus ofrece clientes en varios idiomas. Para obtener más información sobre los clientes, consulte [Prometheus CLIENT LIBRARIES](#). Para obtener más información sobre cómo desarrollar un exportador, consulte [WRITING EXPORTERS](#). La comunidad de Prometheus ofrece varios exportadores externos que pueden ser utilizados directamente. Para obtener más información, consulte [EXPORTERS AND INTEGRATIONS](#).

## Instalación del complemento

Instale el complemento basado en la versión del clúster y los requisitos reales.

- **prometheus**: solo admite los clústeres de v1.21 o anteriores.
- **kube-prometheus-stack**: solo admite los clústeres de v1.23 o posterior. Además de las capacidades de monitorización de prometheus, este complemento proporciona interconexión entre los datos de monitorización y Container Intelligent Analysis (CIA).

## Acceso a Prometheus

Después de instalar el complemento de prometheus, puede desplegar una serie de cargas de trabajo y Services. El StatefulSet de Prometheus se refiere al Prometheus Server.

Puede crear un [LoadBalancer Service](#) de red pública para que se pueda acceder a Prometheus desde una red externa.

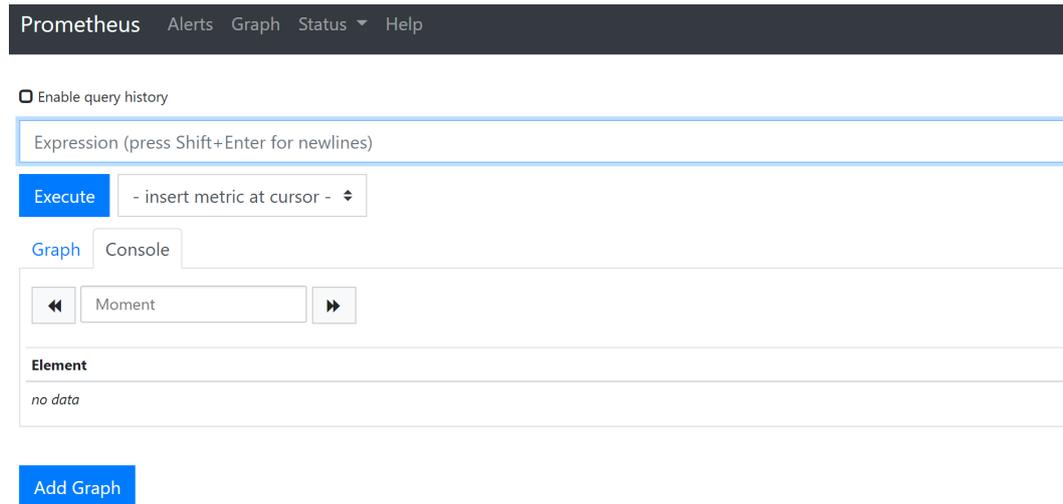
Una vez completada la creación, haga clic en la dirección de acceso para acceder a Prometheus.

**Paso 1** Inicie sesión en la consola de CCE, seleccione un clúster con el complemento de prometheus instalado y elija **Networking** en el panel de navegación.

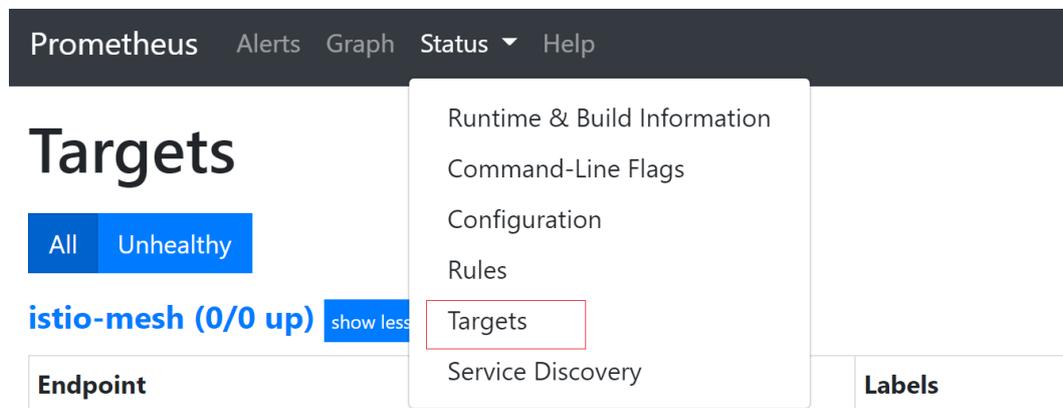
**Paso 2** Haga clic en **Create from YAML** en la esquina superior derecha para crear un Service de LoadBalancer de red pública para la aplicación de Prometheus con estado.

```
apiVersion: v1
kind: Service
metadata:
  name: prom-lb      #Service name, which can be customized.
  namespace: monitoring
  labels:
    app: prometheus
    component: server
  annotations:
    kubernetes.io/elb.id: 038ff***      #Replace it with the ID of the public
networkload balancer in the VPC to which the cluster belongs.
spec:
  ports:
    - name: cce-service-0
      protocol: TCP
      port: 88      #Service port, which can be customized.
      targetPort: 9090      #Default port of Prometheus. Retain the default value.
  selector:
    app: prometheus
    component: server
    release: cceaddon-prometheus
  type: LoadBalancer
```

**Paso 3** Después de la creación, visite *load balancer public IP:Service port* para acceder a Prometheus.



**Paso 4** Elija **Status > Targets** para ver los objetivos monitoreados por Prometheus.



----Fin

## Preparación de una aplicación

Esta sección utiliza Nginx como ejemplo para describir cómo recopilar datos de monitoreo. Las aplicaciones autodesarrolladas necesitan proporcionar API de monitoreo de métricas para que Prometheus pueda invocar. Para obtener más información, véase [Recopilación de datos de monitorización de Prometheus](#).

Nginx tiene un módulo llamado **ngx\_http\_stub\_status\_module** que proporciona funciones básicas de monitorización. Puede configurar el archivo **nginx.conf** para proporcionar una API para que los sistemas externos accedan a los datos de supervisión de Nginx. Como se muestra en la siguiente figura, después de agregar la configuración del servidor a **http**, Nginx puede proporcionar una API para que los sistemas externos accedan a los datos de monitorización de Nginx.

```
user nginx;
worker_processes auto;

error_log /var/log/nginx/error.log warn;
pid /var/run/nginx.pid;

events {
    worker_connections 1024;
```

```
}  
  
http {  
    include          /etc/nginx/mime.types;  
    default_type    application/octet-stream;  
    log_format      main '$remote_addr - $remote_user [$time_local] "$request" ' '  
                        '$status $body_bytes_sent "$http_referer" ' '  
                        '"$http_user_agent" "$http_x_forwarded_for"';  
  
    access_log      /var/log/nginx/access.log  main;  
    sendfile        on;  
    #tcp_nopush     on;  
    keepalive_timeout 65;  
    #gzip           on;  
    include         /etc/nginx/conf.d/*.conf;  
  
    server {  
        listen 8080;  
        server_name localhost;  
        location /stub_status {  
            stub_status on;  
            access_log off;  
        }  
    }  
}
```

Guarde la configuración anterior en el archivo **nginx.conf** y utilice la configuración para crear una nueva imagen. El archivo Dockerfile es el siguiente:

```
FROM nginx:1.21.5-alpine  
ADD nginx.conf /etc/nginx/nginx.conf  
EXPOSE 80  
CMD ["nginx", "-g", "daemon off;"]
```

Utilice el archivo Dockerfile anterior para crear una imagen y subirla a SWR. El nombre de la imagen es **nginx:exporter**. Para obtener más información sobre cómo cargar una imagen, consulte [Carga de una imagen a través de un cliente de motor de contenedores](#).

**docker build -t nginx:exporter .**

**docker tag nginx:exporter swr.ap-southeast-1.myhuaweicloud.com/dev-container/nginx:exporter**

**docker push swr.ap-southeast-1.myhuaweicloud.com/dev-container/nginx:exporter**

Después de ejecutar un contenedor con **nginx:exporter** de imagen, puede obtener datos de monitorización de Nginx llamando a `http://<ip_address> 8080/stub_status`. < ip\_address > indica la dirección IP del contenedor. Los datos de seguimiento son los siguientes:

```
# curl http://127.0.0.1:8080/stub_status  
Active connections: 3  
server accepts handled requests  
146269 146269 212  
Reading: 0 Writing: 1 Waiting: 2
```

## Métricas de supervisión personalizadas

El formato de datos de los datos de seguimiento proporcionados por **nginx:exporter** no cumple con los requisitos de Prometheus. Es necesario convertir el formato de datos al formato requerido por Prometheus. Para convertir el formato de las métricas de Nginx, utilice **nginx-prometheus-exporter**. Despliegue **nginx:exporter** y **nginx-prometheus-exporter** en el mismo pod y agregue las siguientes anotaciones durante el despliegue. Entonces Prometheus puede recopilar automáticamente las métricas.

```
kind: Deployment  
apiVersion: apps/v1
```

```

metadata:
  name: nginx-exporter
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx-exporter
  template:
    metadata:
      labels:
        app: nginx-exporter
      annotations:
        prometheus.io/scrape: "true"
        prometheus.io/port: "9113"
        prometheus.io/path: "/metrics"
        prometheus.io/scheme: "http"
    spec:
      containers:
        - name: container-0
          image: 'nginx:exporter' # Replace it with the address of the image
you uploaded to SWR.
          resources:
            limits:
              cpu: 250m
              memory: 512Mi
            requests:
              cpu: 250m
              memory: 512Mi
        - name: container-1
          image: 'nginx/nginx-prometheus-exporter:0.9.0'
          command:
            - nginx-prometheus-exporter
          args:
            - '-nginx.scrape-uri=http://127.0.0.1:8080/stub_status'
      imagePullSecrets:
        - name: default-secret

```

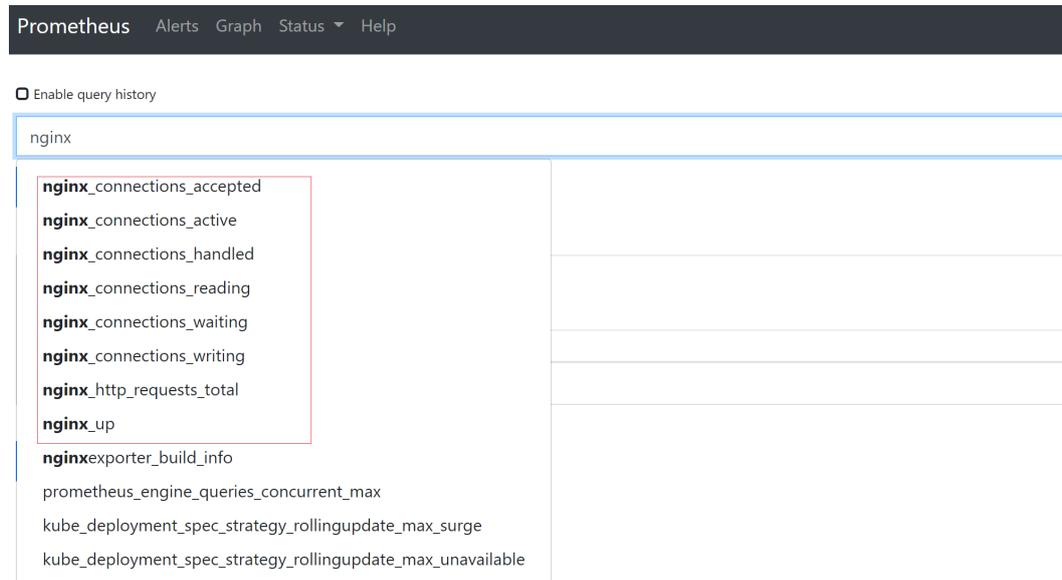
En la descripción anterior:

- **prometheus.io/scrape** indica si se debe permitir que Prometheus recopile datos de monitorización de pods. El valor es **true**.
- **prometheus.io/port** indica el puerto para recopilar datos de monitorización.
- **prometheus.io/path** indica el URL de la API para recopilar datos de supervisión. Si este parámetro no está definido, se utiliza el valor predeterminado **/metrics**.
- **prometheus.io/scheme**: protocolo utilizado para la recogida de datos. El valor puede ser **http** o **https**.

Después de desplegar la aplicación, un pod con una ruta de recopilación del puerto 9113 se puede encontrar en **Status > Targets**.

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="http://10.0.0.133:8080/metrics">http://10.0.0.133:8080/metrics</a>	UP	cluster="15d748b4-3de1-11ec-9199-0255ac1000c9" instance="10.0.0.133:8080" job="kubernetes-pods" kubernetes_namespace="monitoring" kubernetes_pod="cceaddon-prometheus-kube-state-metrics-66dcbd49b-2qhmh"	7.047s ago	4.989ms	
<a href="http://10.0.0.141:9113/metrics">http://10.0.0.141:9113/metrics</a>	UP	cluster="15d748b4-3de1-11ec-9199-0255ac1000c9" instance="10.0.0.141:9113" job="kubernetes-pods" kubernetes_namespace="default" kubernetes_pod="nginx-exporter-33cb99f7/b-b9qwm"	12.914s ago	6.639ms	
<a href="http://10.0.0.7:8080/metrics">http://10.0.0.7:8080/metrics</a>	UP	cluster="15d748b4-3de1-11ec-9199-0255ac1000c9" instance="10.0.0.7:8080" job="kubernetes-pods" kubernetes_namespace="monitoring" kubernetes_pod="cceaddon-prometheus-operator-57acc5bf84-jncitb"	2.156s ago	1.536ms	
<a href="http://10.0.0.8:9090/metrics">http://10.0.0.8:9090/metrics</a>	UP	cluster="15d748b4-3de1-11ec-9199-0255ac1000c9" instance="10.0.0.8:9090" job="kubernetes-pods" kubernetes_namespace="monitoring" kubernetes_pod="prometheus-0"	5.283s ago	5.402ms	

En la página de la ficha **Graph**, escriba **nginx**. Las métricas relacionadas con Nginx se muestran en Prometheus.



## Acceso a Grafana

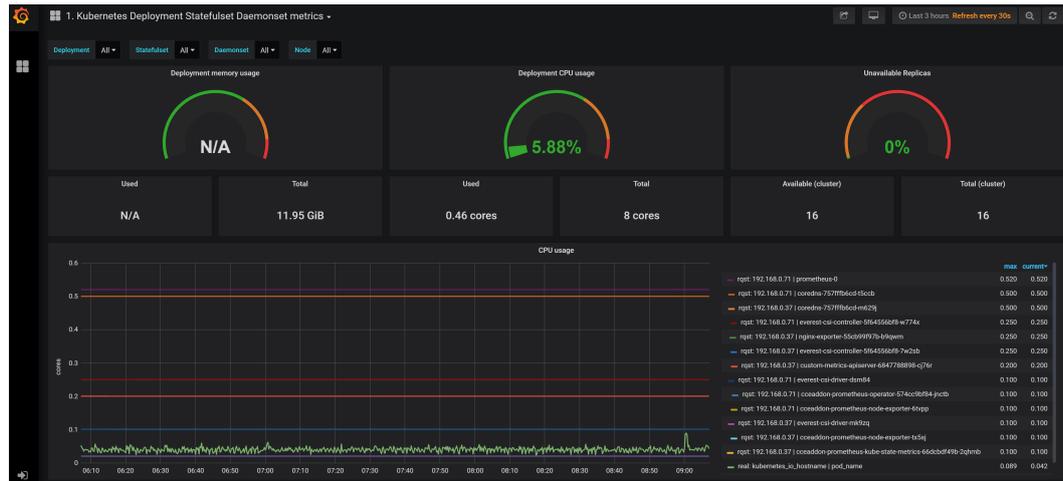
El complemento de prometheus tiene **Grafana** (una herramienta de visualización de código abierto) instalada e interconectada con Prometheus. Puede crear un **Service de LoadBalancer** de red pública para que pueda acceder a Grafana desde la red pública y ver los datos de monitorización de Prometheus en Grafana.

Haga clic en la dirección de acceso para acceder a Grafana y seleccione un panel adecuado para ver el contenido agregado.

- Paso 1** Inicie sesión en la consola de CCE, seleccione un clúster con el complemento de prometheus instalado y elija **Networking** en el panel de navegación.
- Paso 2** Haga clic en **Create from YAML** en la esquina superior derecha para crear un Service de red pública de LoadBalancer para Grafana.

```
apiVersion: v1
kind: Service
metadata:
  name: grafana-lb #Service name, which can be customized.
  namespace: monitoring
  labels:
    app: grafana
  annotations:
    kubernetes.io/elb.id: 038ff*** #Replace it with the ID of the public
networkload balancer in the VPC to which the cluster belongs.
spec:
  ports:
    - name: cce-service-0
      protocol: TCP
      port: 80 #Service port, which can be customized.
      targetPort: 3000 #Default port of Grafana. Retain the default value.
  selector:
    app: grafana
  type: LoadBalancer
```

- Paso 3** Después de la creación, visite **load balancer public IP:Service port** para acceder a Grafana y seleccione un panel adecuado para ver los datos agregados.



----Fin

## Persistencia de los datos de Grafana

Actualmente, los datos de Grafana en el complemento de prometheus no son persistentes. Si se reinicia el contenedor de Grafana, se perderán los datos. Puede montar el almacenamiento en la nube en el contenedor de Grafana para lograr la persistencia de los datos de Grafana.

**Paso 1** Utilice kubectl para conectarse al clúster donde reside el clúster de Grafana. Para obtener más información, véase [Conexión a un clúster con kubectl](#).

**Paso 2** Cree el PVC de un disco de EVS.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: grafana-pvc
  namespace: monitoring
  annotations:
    everest.io/disk-volume-type: SSD
  labels:
    failure-domain.beta.kubernetes.io/region: ap-southeast-1
    failure-domain.beta.kubernetes.io/zone: ap-southeast-1a
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
  storageClassName: csi-disk
```

El disco de EVS y el nodo donde reside Grafana deben estar en la misma AZ. De lo contrario, el disco de EVS no se puede conectar.

- **failure-domain.beta.kubernetes.io/region**: región donde reside el disco de EVS.
- **failure-domain.beta.kubernetes.io/zone**: AZ donde reside el disco de EVS.
- **storage**: tamaño del disco de EVS. Configure este parámetro según sea necesario.

También puede crear discos de EVS en la consola de CCE. Para obtener más información, véase [\(Consola\) Creación automática de un disco de EVS](#).

**Paso 3** Modifique la configuración de la carga de trabajo de Grafana y monte el disco de EVS.

**kubectl edit deployment grafana -n monitoring**

Agregue el disco de EVS al contenedor en el archivo YAML, como se muestra en la siguiente figura. El nombre de PVC debe ser el mismo que el de [Paso 2](#) y la ruta de montaje debe ser `/var/lib/grafana`.

Además, la política de actualización debe modificarse para la carga de trabajo de Grafana. El número máximo de pods es 1.

```
...
  template:
    spec:
      volumes:
        - name: cce-pvc-grafana
          persistentVolumeClaim:
            claimName: grafana-pvc
...
      containers:
        - volumeMounts:
            - name: cce-pvc-grafana
              mountPath: /var/lib/grafana
...
      strategy:
        type: RollingUpdate
        rollingUpdate:
          maxUnavailable: 1
          maxSurge: 1
```

Guarde la configuración. Se actualizará la carga de trabajo de Grafana y se montará el disco de EVS.

---Fin

## 9.4 Monitorización de las métricas del componente del nodo principal

El complemento de [kube-prometheus-stack](#) de 3.5.0 o posterior puede monitorizar los componentes de kube-apiserver, kube-controller, kube-scheduler y etcd-server de los nodos principales. Después de instalar el complemento en un clúster, puede recopilar métricas de los componentes anteriores sin configuración manual.

Esta sección describe cómo compilar el colector Prometheus para recopilar las métricas de los componentes del nodo principal.

### Requisitos previos

- La versión del clúster debe ser 1.19 o posterior.
- Prometheus autoconstruido debe estar instalado en el clúster. Para obtener más información, consulte los [gráficos de Helm de la comunidad de Prometheus](#). El complemento de [prometheus](#) ya no está evolucionado y no es compatible con esta función. Por lo tanto, evite usar este complemento.
- El prometheus-operator debe estar instalado en el clúster. Para obtener más información, consulte [Operador de Prometheus](#).
- Para acceder a Prometheus desde una red externa, cree previamente un [LoadBalancer Service](#) de la red pública para Prometheus.

## Procedimiento

**Paso 1** Utilice **kubectl** para conectarse al clúster.

**Paso 2** Modifique el ClusterRole de Prometheus.

```
kubectl edit ClusterRole prometheus -n {namespace}
```

Agregue el siguiente contenido al campo de reglas:

```
rules:
...
- apiGroups:
  - proxy.exporter.k8s.io
  resources:
  - "*"
  verbs: ["get", "list", "watch"]
```

**Paso 3** Cree un archivo llamado **kube-apiserver.yaml** y edítelo.

```
vi kube-apiserver.yaml
```

Ejemplo de contenido del archivo:

```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  labels:
    app.kubernetes.io/name: apiserver
  name: kube-apiserver
  namespace: monitoring # Change it to the namespace where Prometheus will be
  installed.
spec:
  endpoints:
    - bearerTokenFile: /var/run/secrets/kubernetes.io/serviceaccount/token
      interval: 30s
      metricRelabelings:
        - action: keep
          regex: (aggregator_unavailable_apiservice|
apiserver_admission_controller_admission_duration_seconds_bucket|
apiserver_admission_webhook_admission_duration_seconds_bucket|
apiserver_admission_webhook_admission_duration_seconds_count|
apiserver_client_certificate_expiration_seconds_bucket|
apiserver_client_certificate_expiration_seconds_count|
apiserver_current_inflight_requests|apiserver_request_duration_seconds_bucket|
apiserver_request_total|go_goroutines|kubernetes_build_info|
process_cpu_seconds_total|process_resident_memory_bytes|
rest_client_requests_total|workqueue_adds_total|workqueue_depth|
workqueue_queue_duration_seconds_bucket|aggregator_unavailable_apiservice_total|
rest_client_request_duration_seconds_bucket)
          sourceLabels:
            - __name__
          action: drop
          regex: apiserver_request_duration_seconds_bucket;(0.15|0.25|0.3|0.35|0.4|
0.45|0.6|0.7|0.8|0.9|1.25|1.5|1.75|2.5|3|3.5|4.5|6|7|8|9|15|25|30|50)
          sourceLabels:
            - __name__
            - le
      port: https
      scheme: https
      tlsConfig:
        caFile: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
        serverName: kubernetes
  jobLabel: component
  namespaceSelector:
    matchNames:
      - default
  selector:
    matchLabels:
      component: apiserver
      provider: kubernetes
```

Create a ServiceMonitor:

```
kubectl apply -f kube-apiserver.yaml
```

**Paso 4** Cree un archivo llamado **kube-controller.yaml** y edítelo.

```
vi kube-controller.yaml
```

Ejemplo de contenido del archivo:

```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  labels:
    app.kubernetes.io/name: kube-controller
  name: kube-controller-manager
  namespace: monitoring # Change it to the namespace where Prometheus will be
  installed.
spec:
  endpoints:
    - bearerTokenFile: /var/run/secrets/kubernetes.io/serviceaccount/token
      interval: 15s
      honorLabels: true
      port: https
      relabelings:
        - regex: (.+)
          replacement: /apis/proxy.exporter.k8s.io/v1beta1/kube-controller-proxy/
          sourceLabels:
            - __address__
          targetLabel: __metrics_path__
        - regex: (.+)
          replacement: ${1}
          sourceLabels:
            - __address__
          targetLabel: instance
        - replacement: kubernetes.default.svc.cluster.local:443
          targetLabel: __address__
      scheme: https
      tlsConfig:
        caFile: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
  jobLabel: app
  namespaceSelector:
    matchNames:
      - kube-system
  selector:
    matchLabels:
      app: kube-controller-proxy
      version: v1
```

Cree un ServiceMonitor:

```
kubectl apply -f kube-controller.yaml
```

**Paso 5** Cree un archivo llamado **kube-scheduler.yaml** y edítelo.

```
vi kube-scheduler.yaml
```

Ejemplo de contenido del archivo:

```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  labels:
    app.kubernetes.io/name: kube-scheduler
  name: kube-scheduler
  namespace: monitoring # Change it to the namespace where Prometheus will be
  installed.
spec:
  endpoints:
    - bearerTokenFile: /var/run/secrets/kubernetes.io/serviceaccount/token
      interval: 15s
      honorLabels: true
```

```

    port: https
    relabelings:
      - regex: (.+)
        replacement: /apis/proxy.exporter.k8s.io/v1beta1/kube-scheduler-proxy/
    ${1}/metrics
      sourceLabels:
        - __address__
      targetLabel: __metrics_path__
      - regex: (.+)
        replacement: ${1}
      sourceLabels:
        - __address__
      targetLabel: instance
      - replacement: kubernetes.default.svc.cluster.local:443
        targetLabel: __address__
    scheme: https
    tlsConfig:
      caFile: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
    jobLabel: app
    namespaceSelector:
      matchNames:
        - kube-system
    selector:
      matchLabels:
        app: kube-scheduler-proxy
        version: v1
    
```

Cree un ServiceMonitor:

```
kubectl apply -f kube-scheduler.yaml
```

### Paso 6 Cree un archivo llamado **etcd-server.yaml** y edítelo.

```
vi etcd-server.yaml
```

Ejemplo de contenido del archivo:

```

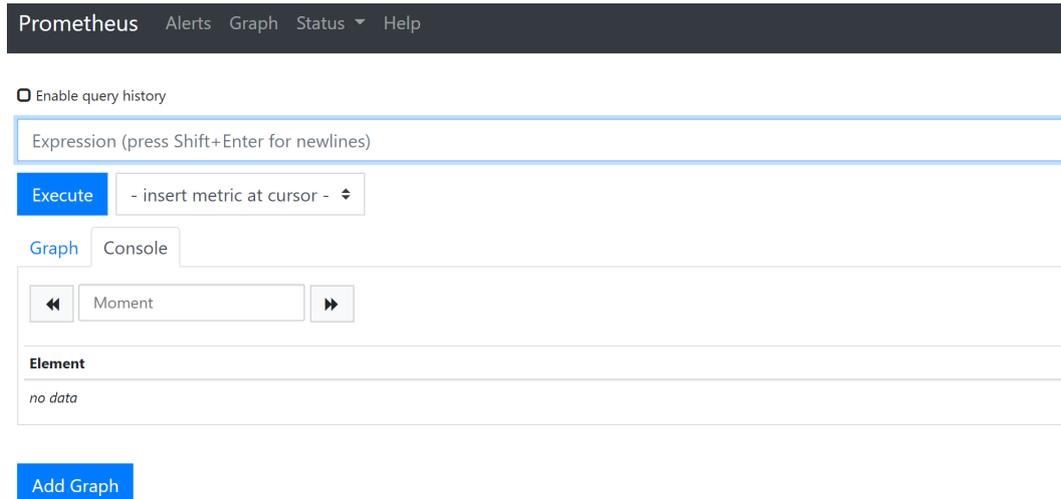
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  labels:
    app.kubernetes.io/name: etcd-server
  name: etcd-server
  namespace: monitoring # Change it to the namespace where Prometheus will be
  installed.
spec:
  endpoints:
    - bearerTokenFile: /var/run/secrets/kubernetes.io/serviceaccount/token
      interval: 15s
      honorLabels: true
      port: https
      relabelings:
        - regex: (.+)
          replacement: /apis/proxy.exporter.k8s.io/v1beta1/etcd-server-proxy/${1}/
  metrics
    sourceLabels:
      - __address__
    targetLabel: __metrics_path__
    - regex: (.+)
      replacement: ${1}
    sourceLabels:
      - __address__
    targetLabel: instance
    - replacement: kubernetes.default.svc.cluster.local:443
      targetLabel: __address__
    scheme: https
    tlsConfig:
      caFile: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
    jobLabel: app
    namespaceSelector:
      matchNames:
    
```

```
- kube-system
selector:
  matchLabels:
    app: etcd-server-proxy
    version: v1
```

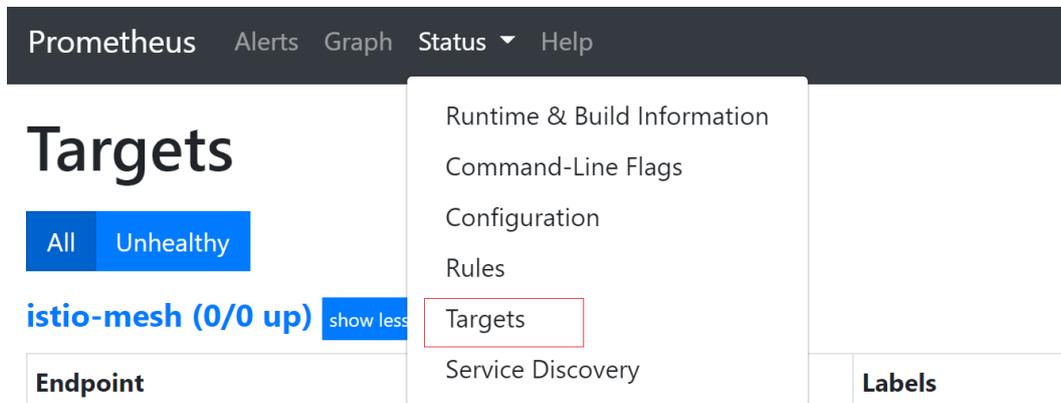
Cree un ServiceMonitor:

```
etcd-server.yaml
```

**Paso 7** Después de la creación, si ha creado un **LoadBalancer Service** de la red pública para Prometheus, puede acceder a **load balancer public IP: Service port** para acceder a Prometheus.



**Paso 8** Elija **Status > Targets**. Se muestran los tres componentes del nodo principal anteriores.



----Fin

## 9.5 Configuraciones de alarma

CCE interactúa con Application Operations Management (AOM) para informar de alarmas y eventos. Al establecer reglas de alarma en AOM, puede comprobar si los recursos de los clústeres son normales de manera oportuna.

## Proceso

1. [Creación de un tema en SMN](#)
2. [Creación de una política de acción](#)
3. Adición de una regla de alarma
  - a. Alarmas de eventos: Genere alarmas basadas en los eventos reportados por clústeres a AOM. Para obtener más información acerca de los eventos y las configuraciones, consulte [Adición de alarmas de eventos](#).
  - b. Alarmas umbral: Genere alarmas basadas en los umbrales de las métricas de monitoreo, como la utilización de recursos de servidores y componentes. Para obtener más información sobre los umbrales y las configuraciones de métricas, consulte [Adición de alarmas de umbral](#).

## Creación de un tema en SMN

Simple Message Notification (SMN) envía mensajes a los suscriptores con correos electrónicos, mensajes SMS y solicitudes HTTP/HTTPS.

Un tema se utiliza para publicar mensajes y suscribirse a notificaciones. Sirve como un canal de transmisión de mensajes entre editores y suscriptores.

Necesita crear un tema y suscribirse a él. Para obtener más información, consulte [Crear un tema](#) y [Suscribirse a un tema](#).

### NOTA

Después de suscribirse a un tema, confirme la suscripción en el correo electrónico o mensaje SMS para que la notificación surta efecto.

## Creación de una política de acción

AOM le permite personalizar las políticas de acción de alarma. Puede crear una política de acción de alarma para asociar un tema de SMN y una plantilla de mensaje. También puede personalizar el contenido de las notificaciones mediante una plantilla de mensaje.

Para obtener más información, consulte [Creación de políticas de acción de alarma](#). Cuando cree una política de acción, seleccione el tema al que se ha creado y al que está suscrito en [Creación de un tema en SMN](#).

## Adición de alarmas de eventos

A continuación se utiliza la alarma **NodeNotReady** como ejemplo para describir cómo agregar una alarma de evento.

Esta función es proporcionada por AOM. Para obtener más información sobre los parámetros, consulte [Creación de reglas de alarma de eventos](#).

**Tabla 9-2** Alarmas basadas en eventos

Nombre del evento	Origen	Descripción	Solución
NodeNotReady	CCE	Una alarma se activa inmediatamente cuando un nodo es anormal.	Inicie sesión en el clúster y compruebe el estado del nodo para el que se genera la alarma. Establezca el nodo como no programado y programe los pods de servicio a otro nodo.
Rebooted	CCE	Una alarma se activa inmediatamente cuando se reinicia un nodo.	Inicie sesión en el clúster para comprobar el estado del nodo para el que se genera la alarma, compruebe si el nodo se puede iniciar correctamente y busque la causa del reinicio.
KUBELETIsDown	CCE	Una alarma se activa inmediatamente cuando un nodo es anormal.	Inicie sesión en el clúster y compruebe el estado del nodo para el que se genera la alarma. Establezca el nodo como no programado y programe los pods de servicio a otro nodo. Luego, reinicie kubelet.
DOCKERIsDown	CCE	Una alarma se activa inmediatamente cuando un nodo es anormal.	Inicie sesión en el clúster y compruebe el estado del nodo para el que se genera la alarma. Establezca el nodo como no programado y programe los pods de servicio a otro nodo. A continuación, reinicie Docker.
KUBEPROXYIsDown	CCE	Una alarma se activa inmediatamente cuando un nodo es anormal.	Inicie sesión en el clúster y compruebe el estado del nodo para el que se genera la alarma. Establezca el nodo como no programado y programe los pods de servicio a otro nodo.
KernelOops	CCE	Una alarma se activa inmediatamente cuando un nodo es anormal.	Inicie sesión en el clúster y compruebe el estado del nodo para el que se genera la alarma. Establezca el nodo como no programado y programe los pods de servicio a otro nodo.
ConntrackFull	CCE	Una alarma se activa inmediatamente cuando un nodo es anormal.	Inicie sesión en el clúster y compruebe el estado del nodo para el que se genera la alarma. Establezca el nodo como no programado y programe los pods de servicio a otro nodo.

Nombre del evento	Origen	Descripción	Solución
NodePoolSoldOut	CCE	Una alarma se activa inmediatamente cuando los recursos del grupo de nodos están agotados.	Establezca la conmutación automática del grupo de nodos o cambie las especificaciones del grupo de nodos.
NodeCreateFailed	CCE	Una alarma se activa inmediatamente después de un fallo de creación de nodo.	Rectifique el error y cree el nodo de nuevo.
ScaleUpTimedOut	CCE	Una alarma se activa inmediatamente después del tiempo de espera de la expansión del nodo.	Rectifique el error e intente expansión de nuevo.
ScaleDownFailed	CCE	Una alarma se activa inmediatamente después del tiempo de espera de la reducción del nodo.	Rectifique el error e intente la reducción de nuevo.
BackOffPullImage	CCE	Error en el reintento de extracción de imagen.	Inicie sesión en el clúster, localice la causa del error y vuelva a desplegar la carga de trabajo del servicio.

**Paso 1** Inicie sesión en la consola de AOM.

**Paso 2** En el panel de navegación, elija **Alarm Center** > **Alarm Rules** y haga clic en **Add Alarm**.

**Paso 3** Establezca una regla de alarma.

- **Rule Type:** Seleccione **Event alarm**.
- **Alarm Source:** Seleccione **CCE**.
- **Select Object:** Seleccione **Event Name** y, a continuación, haga clic en **NodeNotReady**. Puede filtrar objetos activados por tipo de notificación, nombre de evento, gravedad de alarma, atributo personalizado, espacio de nombres y nombre de clúster.
- **Triggering Policy:** Seleccione **Immediate Triggering**.
- **Alarm Mode:** Seleccione **Direct Alarm Reporting**.

- **Action Policy:** seleccione la política de acción creada en [Creación de una política de acción](#).

Esta regla de alarma funciona de la siguiente manera:

Si un nodo en el clúster se vuelve anormal, CCE informa del evento **NodeNotReady** a AOM. AOM le notifica inmediatamente con SMN según la política de acción.

**Figura 9-1** Creación de una alarma de evento

Alarm Rule Settings

Rule type: Threshold Rule | **Event alarms**

\* Alarm Source: CCE

\* Trigger Object:

\* Triggering Policy: Triggering Mode: Immediate Trig... | Alarm Policy: Monitoring per... Accumulated Times >=  times, trigger action strategy

---

Alarm Notification

Alarm Mode: **Direct Alarm Reporting** | Alarm Noise Reduction

Action Policy:  cluster

**Paso 4** Haga clic en **Create Now**.

Si se muestra la siguiente información en la lista de reglas, la regla se crea correctamente.

Alarm Name	Status	Rule Type	Resource Type	Template	Started or Stopped	Operation
NodeNotReady	Effective	Event alarms	CCE	N/A	Started	Modify   Delete   More

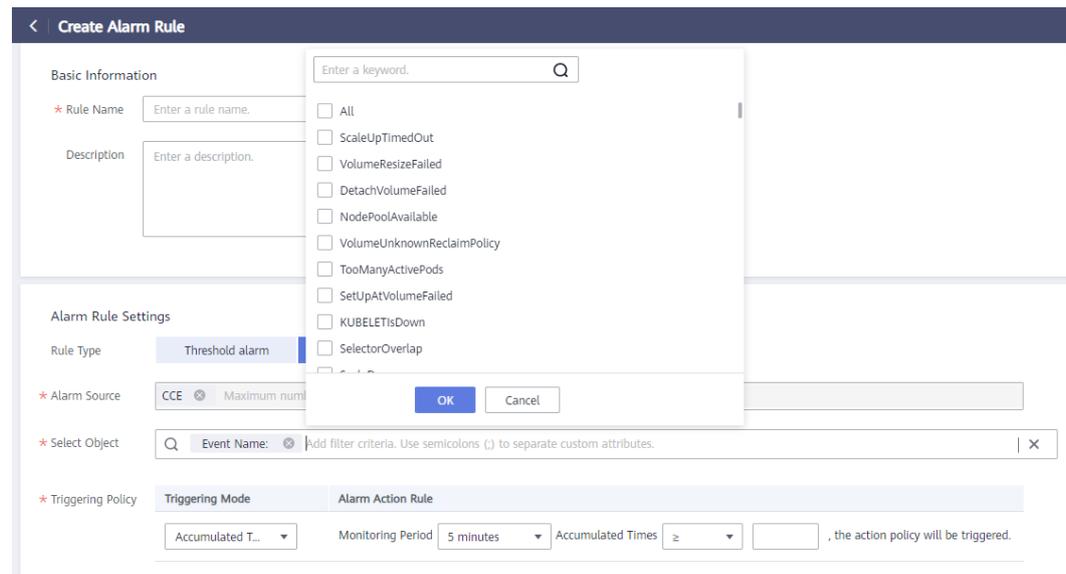
Name	Status	Alarm Source	Trigger Object	Triggering Policy
NodeNotReady	Effective	CCE	Event Name:NodeNotReady;	During the monitoring period:minutesWithin, Ao

----Fin

## Eventos de CCE

Las alarmas de eventos se generan basándose en los eventos notificados por CCE a AOM. CCE informa de una serie de eventos a AOM. Puede ver eventos específicos en las áreas **Alarm Rule Settings** y agregar alarmas de eventos según sea necesario.

**Figura 9-2** Eventos reportados por CCE



CCE admite los siguientes eventos:

- ScaleUpTimedOut
- VolumeResizeFailed
- DetachVolumeFailed
- NodePoolAvailable
- VolumeUnknownReclaimPolicy
- TooManyActivePods
- SetUpAtVolumeFailed
- KUBELETIsDown
- SelectorOverlap
- ScaleDown
- NodeHasInsufficientMemory
- ClaimLost
- UnregisterNetDevice
- VolumeFailedRecycle
- NotTriggerScaleUp
- DeleteUnregistered
- Unhealthy
- FailedDelete
- NetworkCardNotFound
- TooManySucceededPods
- ScaleDownEmpty
- ErrImageNeverPull
- Rebooted
- KUBEPROXYIsDown

- FailedScaleOut
- NodeOutOfDisk
- TaskHung
- WaitForAttachVolumeFailed
- FailedStart
- FailedPullImage
- DeleteNodeWithNoServer
- ReplicaSetCreateError
- CIDRNotAvailable
- ContrackFull
- NodeHasDiskPressure
- FailedStandBy
- ScaleDownFailed
- NodeNotSchedulable
- FailedToScaleUpGroup
- FailedReconfig
- ScaledUpGroup
- NodeInstallFailed
- CreatingLoadBalancerFailed
- FailedGet
- VolumeFailedDelete
- KernelOops
- ScaleUpFailed
- MountDeviceFailed
- DeletingLoadBalancerFailed
- FixNodeGroupSizeDone
- TearDownAtVolumeFailed
- FailedActive
- OOMKilling
- UnmountDeviceFailed
- DOCKERIsDown
- FailedRollback
- CIDRAssignmentFailed
- DockerHung
- SelectingAll
- NodeNotReady
- ProvisioningFailed
- ProvisioningCleanupFailed
- NodeGroupInBackOff
- BackOffStart

- DeploymentRollbackRevisionNotFound
- FailedScheduling
- FixNodeGroupSizeError
- FilesystemIsReadOnly
- FailedUpdate
- NTPsDown
- NodeCreateFailed
- BackOffPullImage
- NodeUninstallFailed
- ClaimMisbound
- FailedList
- NodePoolSoldOut
- AUFSUmountHung
- FailedCreate
- UpdateLoadBalancerFailed
- UnexpectedJob
- FailedScaleIn
- TriggeredScaleUp
- AttachVolumeFailed
- FailedRestart
- CNIsDown
- StartScaledUpGroup
- StartScaleDownEmpty
- DeleteUnregisteredFailed
- Internal error
- External dependency error
- Error al inicializar el subproceso de proceso
- Error al actualizar la base de datos
- Error al crear nodo por grupo de nodos
- Error al eliminar nodo por grupo de nodos
- Error al crear un nodo de suscripción anual/mensual
- Error al cancelar la autorización de acceso a la imagen del principal
- Error al crear la IP virtual para el principal
- Error al eliminar la VM del nodo
- Error al eliminar el grupo de seguridad del nodo
- Error al eliminar el grupo de seguridad del principal
- Error al eliminar el grupo de seguridad del puerto
- Error al eliminar el grupo de seguridad de eni o subeni
- Error al desconectar el puerto del principal
- Error al eliminar el puerto del principal

- Error al eliminar la máquina virtual principal
- Error al eliminar el par de claves del principal
- Error al eliminar la subred del principal
- Error al eliminar la VPC del principal
- Error al eliminar el certificado del clúster
- Error al eliminar el grupo de servidores del principal
- Error al eliminar la IP virtual del principal
- Error al obtener la dirección IP flotante del principal
- Error al obtener la variante de clúster
- Error al obtener el punto del clúster de conexión
- Error al obtener la conexión de Kubernetes
- Error al actualizar el secreto
- Fin del tiempo de espera para la operación
- Se ha agotado el tiempo de espera de la conexión al clúster de Kubernetes
- Error al comprobar el estado del componente o los componentes son anormales
- El nodo no se encuentra en el clúster kubernetes
- El estado del nodo no está listo en el clúster kubernetes
- No se puede encontrar el vm correspondiente de este nodo en ECS
- Error al actualizar el principal
- Error al actualizar el nodo
- Error al cambiar la variante del principal
- Cambiar la variante del tiempo de espera del principal
- Error al pasar la verificación al crear un nodo de suscripción anual/mensual
- Error al instalar el nodo
- Error al limpiar las rutas de la red contenedor del clúster en la VPC
- El estado del clúster es No disponible
- El estado del clúster es un error
- El estado del clúster no se actualiza durante mucho tiempo
- Error al actualizar el estado del principal después de actualizar el tiempo de espera del clúster
- Error al actualizar trabajos en ejecución después de actualizar el tiempo de espera del clúster
- Error al actualizar el estado del clúster
- Error al actualizar el estado del nodo
- Error al quitar el nodo estático de la base de datos
- Error al actualizar el estado del nodo a anormal después del tiempo de espera de procesamiento del nodo
- Error al actualizar el punto de conexión del clúster
- Error al eliminar la conexión no disponible del clúster de Kubernetes.
- Error al sincronizar el certificado de clúster

## Adición de alarmas de umbral

A continuación se utiliza la alarma **Workload CPU Usage** como ejemplo para describir cómo agregar una alarma basada en umbral. También puede utilizar este método para agregar otras alarmas de umbral.

Esta función es proporcionada por AOM. Para obtener más información, consulte [Personalización de reglas de umbral estático](#).

Puede configurar las alarmas de umbral de acuerdo con [Tabla 9-3](#).

### AVISO

El uso de la CPU pod, el uso de la memoria física y las alarmas de uso del sistema de archivos deben configurarse para los componentes everest-csi-controller, everest-csi-driver, coredns, autoscaler y Yangtse. Actualice las especificaciones en el caso de un alto uso de recursos para evitar fallos del sistema.

**Tabla 9-3** Ajustes de alarma umbral

Recurso	Elemento de monitoreo	Descripción	Activador recomendado
Clúster	Uso de CPU	Esta métrica se utiliza para calcular el uso de la CPU del objeto medido.	Condición umbral: > 85%; periodo estadístico (minutos): 1; periodos consecutivos: 3
	Uso de disco	Esta métrica se utiliza para calcular el porcentaje del espacio en disco en uso con respecto al espacio total en disco.	Condición umbral: > 85%; periodo estadístico (minutos): 1; periodos consecutivos: 3
	Uso de la memoria física	Esta métrica se utiliza para calcular el porcentaje de la memoria física utilizada por el objeto medido con respecto a la memoria física total.	Condición umbral: > 85%; periodo estadístico (minutos): 1; periodos consecutivos: 3
	Uso de memoria virtual	Esta métrica se utiliza para calcular el porcentaje de la memoria virtual utilizada por el objeto medido con respecto a la memoria virtual total.	Condición umbral: > 85%; periodo estadístico (minutos): 1; periodos consecutivos: 3
Host	Uso de CPU	Esta métrica se utiliza para calcular el uso de la CPU del objeto medido.	Condición umbral: > 85%; periodo estadístico (minutos): 1; periodos consecutivos: 3

Recurso	Elemento de monitoreo	Descripción	Activador recomendado
	Uso de la memoria física	Esta métrica se utiliza para calcular el porcentaje de la memoria física utilizada por el objeto medido con respecto a la memoria física total.	Condición umbral: > 85%; periodo estadístico (minutos): 1; periodos consecutivos: 3
	Uso de memoria virtual	Esta métrica se utiliza para calcular el porcentaje de la memoria virtual utilizada por el objeto medido con respecto a la memoria virtual total.	Condición umbral: > 85%; periodo estadístico (minutos): 1; periodos consecutivos: 3
Red de host	Tasa de paquetes de errores recibidos	Esta métrica se utiliza para calcular el número de paquetes de error recibidos por una NIC por segundo.	Condición umbral: > 0; periodo estadístico (minutos): 1; periodos consecutivos: 3
	Tasa de paquetes de error de envío	Esta métrica se utiliza para calcular el número de paquetes de error enviados por una NIC por segundo.	Condición umbral: > 0; periodo estadístico (minutos): 1; periodos consecutivos: 3
Sistema de archivos de host	Uso de disco	Esta métrica se utiliza para calcular el porcentaje del espacio en disco en uso con respecto al espacio total en disco.	Condición umbral: > 85%; periodo estadístico (minutos): 1; periodos consecutivos: 3
	Estado de lectura/escritura del disco	Esta métrica se utiliza para recopilar estadísticas sobre el estado de lectura y escritura de discos en un host.	Condición umbral: >= 1; periodo estadístico (minutos): 1; periodos consecutivos: 1
Carga de trabajo	Estado de la carga de trabajo	Esta métrica se utiliza para comprobar el estado de la carga de trabajo anormal.	Condición umbral: >= 1; periodo estadístico (minutos): 1; periodos consecutivos: 1
	Uso de CPU	Esta métrica se usa para calcular el uso de la CPU del objeto medido, a saber, la relación de los núcleos de la CPU usados realmente por el objeto medido con respecto al total de núcleos de la CPU a los que el objeto medido ha aplicado.	Condición umbral: > 85%; periodo estadístico (minutos): 1; periodos consecutivos: 3

Recurso	Elemento de monitoreo	Descripción	Activador recomendado
	Uso de la memoria física	Esta métrica se utiliza para calcular el porcentaje de la memoria física utilizada por el objeto medido con respecto a la memoria física total.	Condición umbral: > 85%; periodo estadístico (minutos): 1; periodos consecutivos: 3
	Uso del sistema de archivos	Esta métrica se utiliza para calcular el uso del sistema de archivos de un objeto medido, es decir, el porcentaje del sistema de archivos utilizado respecto al sistema de archivos total. Esta métrica solo se admite para contenedores mediante Device Mapper en el clúster de Kubernetes de la versión 1.11 o posterior.	Condición umbral: > 85%; periodo estadístico (minutos): 1; periodos consecutivos: 3
Pod	Uso de CPU	Esta métrica se usa para calcular el uso de la CPU del objeto medido, a saber, la relación de los núcleos de la CPU usados realmente por el objeto medido con respecto al total de núcleos de la CPU a los que el objeto medido ha aplicado.	Condición umbral: > 85%; periodo estadístico (minutos): 1; periodos consecutivos: 3
	Uso del sistema de archivos	Esta métrica se utiliza para calcular el uso del sistema de archivos de un objeto medido, es decir, el porcentaje del sistema de archivos utilizado respecto al sistema de archivos total. Esta métrica solo se admite para contenedores mediante Device Mapper en el clúster de Kubernetes de la versión 1.11 o posterior.	Condición umbral: > 85%; periodo estadístico (minutos): 1; periodos consecutivos: 3
	Uso de la memoria física	Esta métrica se utiliza para calcular el porcentaje de la memoria física utilizada por el objeto medido con respecto a la memoria física total.	Condición umbral: > 85%; periodo estadístico (minutos): 1; periodos consecutivos: 3
	Estado del contenedor	Esta métrica se utiliza para comprobar si el estado del contenedor de Docker es normal.	Condición umbral: >= 1; periodo estadístico (minutos): 1; periodos consecutivos: 1

Recurso	Elemento de monitoreo	Descripción	Activador recomendado
	Tasa de paquetes de errores recibidos	Esta métrica se utiliza para calcular el número de paquetes de error recibidos por una NIC por segundo.	Condición umbral: > 0; período estadístico (minutos): 1; periodos consecutivos: 3
	Paquetes recibidos con errores	Esta métrica se utiliza para calcular el número de paquetes de error recibidos por un objeto medido	Condición umbral: > 0; período estadístico (minutos): 1; periodos consecutivos: 3
	Tasa de paquetes de error de envío	Esta métrica se utiliza para calcular el número de paquetes de error enviados por una NIC por segundo.	Condición umbral: > 0; período estadístico (minutos): 1; periodos consecutivos: 3

**Paso 1** Inicie sesión en la consola de AOM.

**Paso 2** En el panel de navegación, elija **Alarm Center** > **Alarm Rules** y haga clic en **Add Alarm**.

**Paso 3** Establezca una regla de alarma.

- **Rule Type:** Seleccione **Threshold Rule**.
- **Monitored Object:** haga clic en **Select resource objects**, establezca **Add By** a **Dimension** y seleccione **CCE/Deployment/CPU Usage** para **Metric Name**. Puede filtrar los recursos por varias dimensiones según sea necesario.

Select Monitored Object

Add By Dimension Resource

Metric Name CCE / Deployment / CPU usage

Dimension Cluster ID:1b2ec02d-a3b2-11ec-b0d0-0255ac100099 Add filter

Confirm Cancel Clear

- **Alarm Condition:** Establezca parámetros como el período estadístico, los tiempos consecutivos y las condiciones de umbral según sea necesario.

\* Alarm Condition Custom Template

Trigger Condition When Monitored Object , during recent 3count monitoring periods, for 1 peri... Avg. > 80

When, the Critical a (xxx) alarm will be generated.

Advanced Settings

- **Triggering Mode:** Seleccione **Immediate Triggering**.
- **Alarm Mode:** Seleccione **Direct Alarm Reporting**.
- **Action Policy:** seleccione la política de acción creada en [Creación de una política de acción](#).

**Paso 4** Haga clic en **Create Now**.

Si se muestra la siguiente información en la lista de reglas, la regla se crea correctamente. En este ejemplo, hay varias cargas de trabajo porque no se especifica ninguna carga de trabajo en los criterios de filtro. Por lo tanto, se muestran todas las cargas de trabajo del clúster.

<input type="checkbox"/>	Alarm Name	Status	Rule Type	Resource Type	Template	Started or Stopped
^	<input type="checkbox"/> workload_cpu_usage	Normal	Multi-resource threshold	Component	N/A	Started
Name	Status	Cluster Name	CPU usage (%)	Status Change Description		
cluster-autoscaler	Insufficient	example	0.097	no metric data		
coredns	Insufficient	example	0.123	no metric data		
everest-csi-controller	Insufficient	example	0.351	no metric data		

----Fin

# 10 Logs

---

## 10.1 Descripción general

CCE le permite configurar políticas para recopilar, gestionar y analizar registros de carga de trabajo periódicamente para evitar que los logs se sobredimensionen.

- Usando ICAgent:

De forma predeterminada, ICAgent recopila las salidas estándar de contenedor (logs de salida). No se requiere configuración.

También puede configurar la ruta para almacenar logs de contenedor al crear una carga de trabajo para que ICAgent recopile logs de esta ruta.

Puede seleccionar cualquiera de los siguientes modos para los logs de contenedor:

- `hostPath`: Se monta una ruta de host en la ruta de contenedor especificada (ruta de montaje). En la ruta del host del nodo, puede ver la salida de los logs de contenedor en la ruta de montaje.
- `emptyDir`: Una ruta temporal del nodo se monta en la ruta especificada (ruta de montaje). Los datos que existen en la ruta temporal pero que el recopilador no notifica a AOM desaparecerán después de eliminar el pod.

## 10.2 Uso de ICAgent para recopilar logs de contenedores

CCE trabaja con AOM para recopilar logs de carga de trabajo. Al crear un nodo, CCE instala el ICAgent por usted (el DaemonSet llamado **icagent** en el espacio de nombres del kube-system del clúster). Una vez que ICAgent recopila los logs de carga de trabajo y los informa a la AOM, puede ver los logs de carga de trabajo en la consola de CCE o de AOM.

### Notas y restricciones

El ICAgent solo recopila archivos de log de texto **\*.log**, **\*.trace** y **\*.out**.

### Facturación

AOM ofrece una cuota de recopilación de logs gratuita de 500 MB por cada cuenta cada mes. Si se excede la cuota, se le cobrará. Para obtener más información, véase la [Facturación](#). Puede hacer clic en la [consola de AOM](#) para ver los logs.

## Uso de ICAgent para recopilar logs

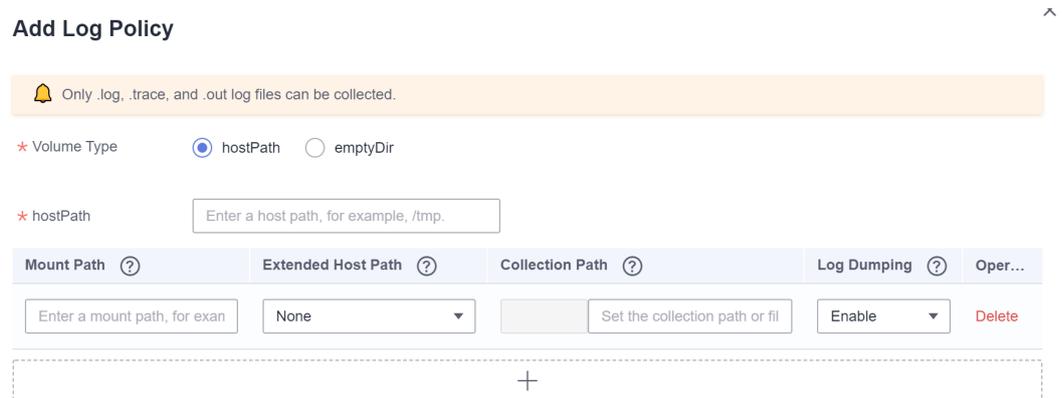
Puede agregar una política para recopilar logs mediante ICAgent para una carga de trabajo.

**Paso 1** Cuando **crea una carga de trabajo**, establezca el registro para el contenedor.

**Paso 2** Haga clic en **+** para agregar una política de log.

Lo siguiente usa Nginx como ejemplo. Las políticas de log se varían en función de las cargas de trabajo.

**Figura 10-1** Adición de una política de log



**Paso 3** Establezca **Storage Type** en **Host Path** o **Container Path**.

**Tabla 10-1** Configuración de políticas de log

Parámetro	Descripción
Storage Type	<ul style="list-style-type: none"> <li>● <b>Host Path</b> (hostPath): Una ruta de host se monta en la ruta de contenedor especificada. En la ruta del host del nodo, puede ver la salida de los logs de contenedor en la ruta de montaje.</li> <li>● <b>Container Path</b> (emptyDir): Una ruta temporal del nodo se monta en la ruta especificada. Los datos que existen en la ruta temporal pero que el recopilador no notifica a AOM desaparecerán después de eliminar el pod.</li> </ul>
Host Path	Introduzca una ruta de acceso de host, por ejemplo, <b>/var/paas/sys/log/nginx</b> .

Parámetro	Descripción
Container Path	<p>Ruta del contenedor (por ejemplo, <b>/tmp</b>) en la que se montarán los recursos de almacenamiento.</p> <p><b>AVISO</b></p> <ul style="list-style-type: none"> <li>● No monte el almacenamiento en un directorio del sistema como <b>/o/var/run</b>; esta acción puede provocar un error de contenedor. Se recomienda montar el contenedor en un directorio vacío. Si el directorio no está vacío, asegúrese de que no haya archivos que afecten al inicio del contenedor en el directorio. De lo contrario, dichos archivos se reemplazarán, lo que provocará errores al iniciar el contenedor y crear la carga de trabajo.</li> <li>● Cuando el contenedor está montado en un directorio de alto riesgo, se recomienda utilizar una cuenta con los permisos mínimos para iniciar el contenedor; de lo contrario, los archivos de alto riesgo en la máquina host podrían estar dañados.</li> <li>● AOM recopila solo los primeros 20 archivos de log que se han modificado recientemente. Recoge archivos de 2 niveles de subdirectorios de forma predeterminada.</li> <li>● AOM solo recopila los archivos de texto de log <b>.log</b>, <b>.trace</b> y <b>.out</b> en las rutas de montaje.</li> <li>● Para obtener más información acerca de cómo establecer permisos para puntos de montaje en un contenedor, consulte <a href="#">Configurar un contexto de seguridad para un pod o un contenedor</a>.</li> </ul>
Extended Host Path	<p>Este parámetro solo es obligatorio si <b>Storage Type</b> está establecido en <b>Host Path</b>.</p> <p>Las rutas de host extendidas contienen ID de pod o nombres de contenedor para distinguir diferentes contenedores en las que está montada la ruta de host.</p> <p>Se agregue un directorio de nivel 3 al directorio/subdirectorio de volumen original. Puede obtener fácilmente la salida de archivos por un solo Pod.</p> <ul style="list-style-type: none"> <li>● <b>None</b>: No se ha configurado ninguna ruta extendida.</li> <li>● <b>PodUID</b>: ID de un pod.</li> <li>● <b>PodName</b>: Nombre de un pod.</li> <li>● <b>PodUID/ContainerName</b>: ID de un pod o nombre de un contenedor.</li> <li>● <b>PodName/ContainerName</b>: Nombre de un pod o un contenedor.</li> </ul>

Parámetro	Descripción
Ruta de colección	<p>Una ruta de recopilación limita el ámbito de la recopilación a los registros especificados.</p> <ul style="list-style-type: none"> <li>● Si no se especifica ninguna ruta de recopilación, los archivos de log de los formatos <b>.log</b>, <b>.trace</b> y <b>.out</b> se recopilarán a partir de la ruta especificada.</li> <li>● <b>/Path/**/</b> indica que todos los archivos de log de los formatos <b>.log</b>, <b>.trace</b> y <b>.out</b> se recopilarán de forma recursiva desde la ruta especificada y todos los subdirectorios a 5 niveles de profundidad.</li> <li>● <b>*</b> en los nombres de los archivos de log indica una coincidencia difusa.</li> </ul> <p>Ejemplo: La ruta de recopilación <b>/tmp/**/test*.log</b> indica que todos los archivos <b>.log</b> con el prefijo <b>test</b> se recopilarán de <b>/tmp</b> y subdirectorios a 5 niveles de profundidad.</p> <p><b>ATENCIÓN</b>                      Asegúrese de que la versión de <b>ICAgent</b> es 5.12.22 o posterior.</p>
Log Dump	<p>El volcado de logs se refiere a la rotación de archivos de log en un host local.</p> <ul style="list-style-type: none"> <li>● <b>Enabled:</b> AOM analiza los archivos de log cada minuto. Cuando un archivo de log supera los 50 MB, se descarga inmediatamente. Se genera un nuevo archivo <b>.zip</b> en el directorio donde se encuentra el archivo de log. Para un archivo de log, AOM almacena solo los 20 archivos de <b>.zip</b> más recientes. Cuando el número de archivos <b>.zip</b> supera los 20, se eliminarán los archivos <b>.zip</b> anteriores. Una vez completado el volcado, se borrará el archivo de log en AOM.</li> <li>● <b>Disabled:</b> AOM no volca los archivos de log.</li> </ul> <p><b>NOTA</b></p> <ul style="list-style-type: none"> <li>● AOM rota los archivos de log mediante copytruncate. Antes de habilitar el volcado de log, asegúrese de que los archivos de log estén escritos en el modo de adición. De lo contrario, pueden producirse agujeros de archivo.</li> <li>● Actualmente, los componentes principales de log como Log4j y Logback admiten la rotación de archivos de log. Si ya ha establecido la rotación para los archivos de log, omita la configuración. De lo contrario, pueden producirse conflictos.</li> <li>● Se recomienda configurar la rotación de archivos de log para sus propios servicios para controlar de manera flexible el tamaño y el número de archivos enrollados.</li> </ul>

**Paso 4** Haga clic en **OK**.

----Fin

## Ejemplo de YAML (ICAgent)

Puede establecer la ruta de almacenamiento del log contenedor definiendo un archivo YAML.

Como se muestra en la siguiente figura, un `emptyDir` está montado en una ruta temporal a `/var/log/nginx`. De esta manera, el ICAgent recopila los logs de `/var/log/nginx`. El campo **policy** es personalizado por CCE y permite al ICAgent identificar y recopilar logs.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: testlog
  namespace: default
spec:
  selector:
    matchLabels:
      app: testlog
  template:
    replicas: 1
    metadata:
      labels:
        app: testlog
    spec:
      containers:
        - image: 'nginx:alpine'
          name: container-0
          resources:
            requests:
              cpu: 250m
              memory: 512Mi
            limits:
              cpu: 250m
              memory: 512Mi
          volumeMounts:
            - name: vol-log
              mountPath: /var/log/nginx
              policy:
                logs:
                  rotate: ''
      volumes:
        - emptyDir: {}
          name: vol-log
    imagePullSecrets:
      - name: default-secret
```

A continuación se muestra cómo utilizar un volumen de `hostPath`. En comparación con `emptyDir`, el tipo de **volumes** se cambia a **hostPath** y la ruta en el host debe configurarse para este volumen de `hostPath`. En el siguiente ejemplo, el `/tmp/log` del host se monta en `/var/log/nginx`. De esta manera, el ICAgent puede recopilar los logs de `/var/log/nginx` sin eliminar los logs de `/tmp/log`.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: testlog
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      app: testlog
  template:
    metadata:
      labels:
        app: testlog
    spec:
      containers:
        - image: 'nginx:alpine'
          name: container-0
          resources:
            requests:
              cpu: 250m
```

```

        memory: 512Mi
        limits:
          cpu: 250m
          memory: 512Mi
      volumeMounts:
      - name: vol-log
        mountPath: /var/log/nginx
        readOnly: false
        extendPathMode: PodUID
        policy:
          logs:
            rotate: Hourly
            annotations:
              pathPattern: '*'
              format: ''
      volumes:
      - hostPath:
          path: /tmp/log
          name: vol-log
      imagePullSecrets:
      - name: default-secret
    
```

**Tabla 10-2** Descripción de parámetros

Parámetro	Descripción	Descripción
extendPathMode	Ruta de host extendida	<p>Las rutas de host extendidas contienen ID de pod o nombres de contenedor para distinguir diferentes contenedores en las que está montada la ruta de host.</p> <p>Se agregue un directorio de nivel 3 al directorio/subdirectorio de volumen original. Puede obtener fácilmente la salida de archivos por un solo Pod.</p> <ul style="list-style-type: none"> <li>● <b>None</b>: No se ha configurado ninguna ruta extendida.</li> <li>● <b>PodUID</b>: ID de un pod.</li> <li>● <b>PodName</b>: Nombre de un pod.</li> <li>● <b>PodUID/ContainerName</b>: ID de un pod o nombre de un contenedor.</li> <li>● <b>PodName/ContainerName</b>: Nombre de un pod o un contenedor.</li> </ul>

Parámetro	Descripción	Descripción
policy.logs.rotate	Volcado de logs	<p>El volcado de logs se refiere a la rotación de archivos de log en un host local.</p> <ul style="list-style-type: none"> <li>● <b>Enabled:</b> AOM analiza los archivos de log cada minuto. Cuando un archivo de log supera los 50 MB, se descarga inmediatamente. Se genera un nuevo archivo <b>.zip</b> en el directorio donde se encuentra el archivo de log. Para un archivo de log, AOM almacena solo los 20 archivos de <b>.zip</b> más recientes. Cuando el número de archivos <b>.zip</b> supera los 20, se eliminarán los archivos <b>.zip</b> anteriores. Una vez completado el volcado, se borrará el archivo de log en AOM.</li> <li>● <b>Disabled:</b> AOM no volca los archivos de log.</li> </ul> <p><b>NOTA</b></p> <ul style="list-style-type: none"> <li>● AOM rota los archivos de log mediante copytruncate. Antes de habilitar el volcado de log, asegúrese de que los archivos de log estén escritos en el modo de adición. De lo contrario, pueden producirse agujeros de archivo.</li> <li>● Actualmente, los componentes principales de log como Log4j y Logback admiten la rotación de archivos de log. Si ha establecido la rotación para los archivos de log, omita la configuración. De lo contrario, pueden producirse conflictos.</li> <li>● Se recomienda configurar la rotación de archivos de log para sus propios servicios para controlar de manera flexible el tamaño y el número de archivos enrollados.</li> </ul>
policy.logs.annotationPath	Ruta de colección	<p>Una ruta de recopilación limita el ámbito de la recopilación a los registros especificados.</p> <ul style="list-style-type: none"> <li>● Si no se especifica ninguna ruta de recopilación, los archivos de log de los formatos <b>.log</b>, <b>.trace</b> y <b>.out</b> se recopilarn a partir de la ruta especificada.</li> <li>● <b>/Path/**/</b> indica que todos los archivos de log de los formatos <b>.log</b>, <b>.trace</b> y <b>.out</b> se recopilarn de forma recursiva desde la ruta especificada y todos los subdirectorios a 5 niveles de profundidad.</li> <li>● <b>*</b> en los nombres de los archivos de log indica una coincidencia difusa.</li> </ul> <p>Ejemplo: La ruta de recopilación <b>/tmp/**/test*.log</b> indica que todos los archivos <b>.log</b> con el prefijo <b>test</b> se recopilarn de <b>/tmp</b> y subdirectorios a 5 niveles de profundidad.</p> <p><b>ATENCIÓN</b>                      Asegúrese de que la versión de <b>ICAgent</b> es 5.12.22 o posterior.</p>

Parámetro	Descripción	Descripción
policy.logs.annotations.format	Coincidencia de logs de varias líneas	<p>Algunos registros de programa (por ejemplo, los logs de programa Java) contienen un log que ocupa varias líneas. De forma predeterminada, el sistema de recopilación de logs recopila logs por línea. Si desea mostrar los logs como un único mensaje de log en el sistema de recopilación de logs, puede habilitar la función de log de varias líneas y usar el modo de log de tiempo o patrón regular. Cuando una línea de mensaje de log coincide con el formato de tiempo preestablecido o la expresión regular, se considera como el inicio de un mensaje de log y la siguiente línea comienza con esta línea de mensaje de log se considera como el identificador de fin del mensaje de log.</p> <p>El formato es el siguiente:</p> <pre>{   "multi": {     "mode": "time",     "value": "YYYY-MM-DD hh:mm:ss"   } }</pre> <p><b>multi</b> indica el modo multilínea.</p> <ul style="list-style-type: none"> <li>● <b>time</b>: tiempo de log. Introduzca un comodín de tiempo. Por ejemplo, si la hora en el log es 2017-01-01 23:59:59, el comodín es AAAA-MM-DD hh:mm:ss.</li> <li>● <b>regular</b>: patrón regular. Ingresar una expresión regular.</li> </ul>

## Visualización de logs

Después de configurar una ruta de recopilación de logs y crear la carga de trabajo, ICAgent recopila los archivos de log de la ruta de acceso configurada. La recolección dura aproximadamente 1 minuto.

Una vez completada la recopilación de logs, vaya a la página de detalles de la carga de trabajo y haga clic en **Logs** en la esquina superior derecha para ver los logs.

También puede ver los logs en la consola de AOM.

También puede ejecutar el comando **kubectl logs** para ver la salida estándar de un contenedor.

```
# View logs of a specified pod.
kubectl logs <pod_name>
kubectl logs -f <pod_name> # Similar to tail -f

# View logs of a specified container in a specified pod.
kubectl logs <pod_name> -c <container_name>

kubectl logs pod_name -c container_name -n namespace (one-off query)
kubectl logs -f <pod_name> -n namespace (real-time query in tail -f mode)
```

# 11 Namespaces

---

## 11.1 Creación de un espacio de nombres

### Cuándo usar espacios de nombres

Un espacio de nombres es una colección de recursos y objetos. Se pueden crear varios espacios de nombres dentro de un clúster y aislarlos unos de otros. Esto permite que los espacios de nombres compartan los mismos servicios de clúster sin afectarse entre sí.

Por ejemplo, puede desplegar cargas de trabajo en un entorno de desarrollo en un espacio de nombres y desplegar cargas de trabajo en un entorno de pruebas en otro espacio de nombres.

### Requisitos previos

Se ha creado al menos un clúster.

### Notas y restricciones

Se puede crear un máximo de 6,000 Services en cada espacio de nombres. Los Services mencionados aquí indican los recursos de Service de Kubernetes agregados para las cargas de trabajo.

### Tipos de espacio de nombres

Los espacios de nombres se pueden crear de cualquiera de las siguientes maneras:

- Creado automáticamente: cuando un clúster está activo, los espacios de nombres **default**, **kube-public**, **kube-system** y **kube-node-lease** se crean de forma predeterminada.
  - **default**: Todos los objetos para los que no se especifica ningún espacio de nombres se asignan a este espacio de nombres.
  - **kube-public**: Todos los usuarios pueden acceder a los recursos de este espacio de nombres (incluidos los usuarios no autenticados), como complementos públicos y gráficos de contenedor.
  - **kube-system**: Todos los recursos creados por Kubernetes se encuentran en este espacio de nombres.

- **kube-node-lease:** Cada nodo tiene un objeto Lease asociado en este espacio de nombres. El objeto es actualizado periódicamente por el nodo. Tanto NodeStatus como NodeLease se consideran latidos del corazón de un nodo. En versiones anteriores a la v1.13, solo NodeStatus está disponible. La función NodeLease se introduce en la v1.13. NodeLease es más ligero que el NodeStatus. Esta característica mejora significativamente la escalabilidad y el rendimiento del clúster.
- **Creado manualmente:** Puede crear espacios de nombres para fines separados. Por ejemplo, puede crear tres espacios de nombres, uno para un entorno de desarrollo, uno para un entorno de depuración conjunta y otro para un entorno de prueba. También puede crear un espacio de nombres para los servicios de inicio de sesión y uno para los servicios de juegos.

## Creación de un espacio de nombres

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** Elija **Namespaces** en el panel de navegación y haga clic en **Create Namespace** en la esquina superior derecha.

**Paso 3** Establezca los parámetros de espacio de nombres según [Tabla 11-1](#).

**Tabla 11-1** Parámetros para crear un espacio de nombres

Parámetro	Descripción
Name	Nombre único del espacio de nombres creado.
Description	Descripción sobre el espacio de nombres.
Quota Management	<p>Las cuotas de recursos pueden limitar la cantidad de recursos disponibles en los espacios de nombres, logrando la asignación de recursos por espacio de nombres.</p> <p><b>AVISO</b></p> <p><b>Se recomienda establecer cuotas de recursos en el espacio de nombres según sea necesario para evitar excepciones de clúster o nodo causadas por la sobrecarga de recursos.</b></p> <p>Por ejemplo, el número predeterminado de pods que se pueden crear en cada nodo en un clúster es 110. Si crea un clúster con 50 nodos, puede crear un máximo de 5,500 pods. Por lo tanto, puede establecer una cuota de recursos para asegurarse de que el número total de pods en todos los espacios de nombres no exceda de 5,500.</p> <p>Escriba un número entero. Si no se especifica la cuota de un recurso, no se plantea ningún límite en el recurso.</p> <p>Si desea limitar la cuota de CPU o memoria, debe especificar el valor de solicitud de CPU o memoria al crear una carga de trabajo.</p>

**Paso 4** Cuando se complete la configuración, haga clic en **OK**.

----Fin

## Uso de kubectl para crear un espacio de nombres

Defina un espacio de nombres.

```
apiVersion: v1
kind: Namespace
metadata:
  name: custom-namespace
```

Ejecute el comando **kubectl** para crearlo.

```
$ kubectl create -f custom-namespace.yaml
namespace/custom-namespace created
```

También puede ejecutar el comando **kubectl create namespace** para crear un espacio de nombres.

```
$ kubectl create namespace custom-namespace
namespace/custom-namespace created
```

## 11.2 Gestión de espacios de nombres

### Uso de espacios de nombres

- Al crear una carga de trabajo, puede seleccionar un espacio de nombres para aislar recursos o usuarios.
- Al consultar cargas de trabajo, puede seleccionar un espacio de nombres para ver todas las cargas de trabajo del espacio de nombres.

### Aislamiento de espacios de nombres

- **Aislamiento de espacios de nombres por entorno**

Una aplicación generalmente pasa por las etapas de desarrollo, depuración conjunta y prueba antes de lanzarse. En este proceso, las cargas de trabajo desplegadas en cada entorno (etapa) son las mismas, pero están definidas lógicamente. Hay dos formas a definirlos:

- Agruparlos en diferentes clústeres para diferentes entornos.

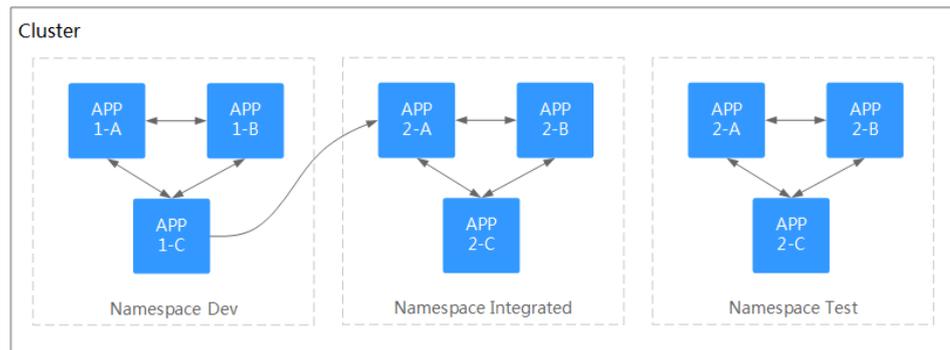
Los recursos no se pueden compartir entre diferentes clústeres. Además, los servicios en diferentes entornos pueden acceder entre sí solo a través del balanceo de carga.

- Agruparlos en diferentes espacios de nombres para diferentes entornos.

Se puede acceder mutuamente a las cargas de trabajo del mismo espacio de nombres mediante el nombre de Service. El acceso entre espacios de nombres se puede implementar mediante el nombre de Service o el nombre de espacio de nombres.

La siguiente figura muestra los espacios de nombres creados para los entornos de desarrollo, depuración conjunta y prueba, respectivamente.

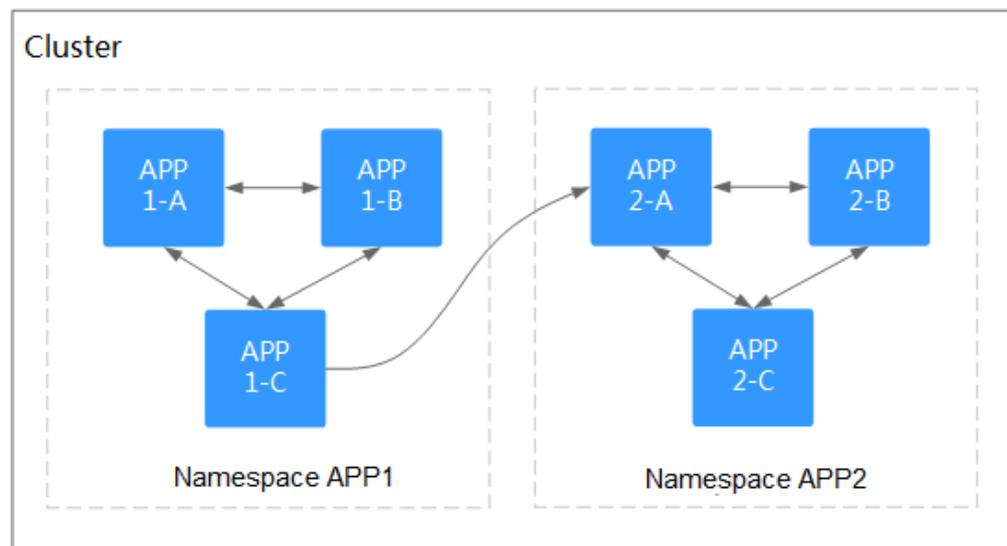
**Figura 11-1** Un espacio de nombres para un entorno



- **Aislamiento de espacios de nombres por aplicación**

Se recomienda utilizar este método si se despliegan un gran número de cargas de trabajo en el mismo entorno. Por ejemplo, en la siguiente figura, se crean diferentes espacios de nombres (APP1 y APP2) para gestionar lógicamente las cargas de trabajo como diferentes grupos. Las cargas de trabajo del mismo espacio de nombres tienen acceso entre sí mediante el nombre de Service y las cargas de trabajo de diferentes espacios de nombres tienen acceso entre sí mediante el nombre de Service o el nombre de espacio de nombres.

**Figura 11-2** Agrupación de cargas de trabajo en diferentes espacios de nombres



## Eliminación de un espacio de nombres

Si se elimina un espacio de nombres, todos los recursos (como cargas de trabajo, trabajos y ConfigMaps) de este espacio de nombres también se eliminarán. Tenga cuidado al eliminar un espacio de nombres.

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** Elija **Namespaces** en el panel de navegación. En la página mostrada, haga clic en **More** en la fila del espacio de nombres de destino y elija **Delete**.

Siga las instrucciones para eliminar el espacio de nombres. No se pueden eliminar los espacios de nombres predeterminados.

---Fin

## 11.3 Establecimiento de una cuota de recursos

Las cuotas de recursos a nivel de espacio de nombres limitan la cantidad de recursos disponibles para los equipos o usuarios cuando estos equipos o usuarios utilizan el mismo clúster. Las cuotas incluyen el número total de un tipo de objetos y la cantidad total de recursos de cálculo (CPU y memoria) consumidos por los objetos.

### Uso

De forma predeterminada, los pods en ejecución pueden usar las CPU y la memoria de un nodo sin restricciones. Esto significa que los pods de un espacio de nombres pueden agotar todos los recursos del clúster.

Kubernetes proporciona espacios de nombres para que pueda agrupar cargas de trabajo en un clúster. Al establecer cuotas de recursos para cada espacio de nombres, puede evitar el agotamiento de recursos y garantizar la fiabilidad del clúster.

Puede configurar cuotas para recursos como la CPU, la memoria y el número de pods en un espacio de nombres. Para obtener más información, consulte [Cuotas de recursos](#).

En la siguiente tabla se recomienda cuántos pods puede configurar para sus clústeres de diferentes tamaños.

Escala de clúster	Número recomendado de pods
50 nodos	2,500 pods
200 nodos	10,000 pods
1,000 nodos	30,000 pods
2,000 nodos	50,000 pods

A partir de clústeres de v1.21 y posteriores, las [Cuotas de recursos](#) predeterminadas se crean cuando se crea un espacio de nombres si ha habilitado **enable-resource-quota** en [Gestión de configuración de clúster](#). **Tabla 11-2** enumera las cuotas de recursos basadas en las especificaciones del clúster. Puede modificarlos de acuerdo con sus requisitos de servicio.

**Tabla 11-2** Cuotas de recursos predeterminadas

Escala de clúster	Pod	Deployment	Secret	ConfigMap	Service
50 nodos	2000	1000	1000	1000	1000
200 nodos	2000	1000	1000	1000	1000
1,000 nodos	5000	2000	2000	2000	2000

Escala de clúster	Pod	Deployment	Secreto	ConfigMap	Service
2,000 nodos	5000	2000	2000	2000	2000

## Restricciones

Kubernetes proporciona control de simultaneidad optimista (OCC), también conocido como bloqueo optimista, para actualizaciones de datos frecuentes. Puede utilizar el bloqueo optimista definiendo el campo **resourceVersion**. Este campo se encuentra en los metadatos del objeto. Este campo identifica el número de versión interno del objeto. Cuando se modifica el objeto, este campo se modifica en consecuencia. Puede usar kube-apiserver para comprobar si un objeto ha sido modificado. Cuando el servidor de API recibe una solicitud de actualización que contiene el campo **resourceVersion**, el servidor compara los datos solicitados con el número de versión de recurso del servidor. Si son diferentes, el objeto del servidor se ha modificado cuando se envía la actualización. En este caso, el servidor de API devuelve un error de conflicto (409). Debe obtener los datos del servidor, modificarlos y enviarlos de nuevo al servidor. La cuota de recursos limita el consumo total de recursos de cada espacio de nombres y registra la información de recursos en el clúster. Por lo tanto, una vez activada la opción **enable-resource-quota**, la probabilidad de conflictos de creación de recursos aumenta en escenarios de simultaneidad a gran escala, lo que afecta al rendimiento de la creación de recursos por lotes.

## Procedimiento

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** En el panel de navegación, haga clic en **Namespaces**.

**Paso 3** Haga clic en **Quota Management** junto al espacio de nombres de destino.

Esta operación no se puede realizar en los espacios de nombres del sistema **kube-system** y **kube-public**.

**Paso 4** Establezca las cuotas de recursos y haga clic en **OK**.

### AVISO

- Después de establecer las cuotas de CPU y memoria para un espacio de nombres, debe especificar los valores de solicitud y límite de recursos de CPU y memoria al crear una carga de trabajo. De lo contrario, no se puede crear la carga de trabajo. Si la cuota de un recurso se establece en **0**, el uso del recurso no está limitado.
- El uso de cuotas acumuladas incluye los recursos utilizados por CCE para crear componentes predeterminados, como los Kubernetes Services (que se pueden ver con kubectl) creados bajo el espacio de nombres **default**. Por lo tanto, se recomienda establecer una cuota de recursos mayor que la esperada para reservar recursos para crear componentes predeterminados.

----Fin

# 12 ConfigMaps y Secretos

---

## 12.1 Creación de un ConfigMap

### Escenario

Un ConfigMap es un tipo de recurso que almacena la información de configuración requerida por una carga de trabajo. Su contenido está definido por el usuario. Después de crear ConfigMaps puede usarlos como archivos o variables de entorno en una carga de trabajo contenedorizada.

ConfigMaps le permite desacoplar archivos de configuración de imágenes de contenedor para mejorar la portabilidad de las cargas de trabajo.

Beneficios de ConfigMaps:

- Gestionar configuraciones de diferentes entornos y servicios.
- Desplegar cargas de trabajo en diferentes entornos. Se admiten varias versiones para los archivos de configuración para que pueda actualizar y revertir fácilmente las cargas de trabajo.
- Importar rápidamente configuraciones en forma de archivos a contenedores.

### Restricciones

- El tamaño de un archivo de recursos de ConfigMap no puede superar los 2 MB.
- ConfigMaps no se puede utilizar en [pods estáticos](#).

### Procedimiento

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** Elija **ConfigMaps and Secrets** en el panel de navegación y haga clic en **Create ConfigMap** en la esquina superior derecha.

**Paso 3** Establezca los parámetros.

**Tabla 12-1** Parámetros para crear un ConfigMap

Parámetro	Descripción
Name	Nombre de un ConfigMap que debe ser único en un espacio de nombres.
Namespace	Espacio de nombres al que pertenece el ConfigMap. Si no especifica este parámetro, el valor <b>default</b> se utiliza de forma predeterminada.
Description	Descripción del ConfigMap.
Data	Datos de un ConfigMap en el formato de par clave-valor. Haga clic en  para agregar datos. El valor puede estar en formato cadena, JSON o YAML.
Label	Etiqueta del ConfigMap. Ingrese un par clave-valor y haga clic en <b>Add</b> .

**Paso 4** Una vez completada la configuración, haga clic en **OK**.

El nuevo ConfigMap se muestra en la lista de ConfigMap.

----Fin

## Creación de un ConfigMap con kubectl

**Paso 1** Configure el comando **kubectl** para conectar un ECS al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).

**Paso 2** Cree un archivo llamado **cce-configmap.yaml** y edítelo.

**vi cce-configmap.yaml**

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: cce-configmap
data:
  SPECIAL_LEVEL: Hello
  SPECIAL_TYPE: CCE
```

**Tabla 12-2** Parámetros de clave

Parámetro	Descripción
apiVersion	El valor se fija en <b>v1</b> .
kind	El valor se fija en <b>ConfigMap</b> .
metadata.name	Nombre de ConfigMap, que se puede personalizar.
data	Datos de ConfigMap. El valor debe ser pares de clave-valor.

**Paso 3** Ejecute los siguientes comandos para crear un ConfigMap.

**kubectl create -f cce-configmap.yaml**

Ejecute los siguientes comandos para ver el ConfigMap creado:

**kubectl get cm**

NAME	DATA	AGE
cce-configmap	3	7m

----Fin

## Operaciones relacionadas

Después de crear un elemento de configuración, puede actualizarlo o eliminarlo como se describe en [Tabla 12-3](#).

**Tabla 12-3** Operaciones relacionadas

Operación	Descripción
Edición de un archivo YAML	Haga clic en <b>Edit YAML</b> en la fila donde reside el ConfigMap de destino para editar su archivo YAML.
Actualización de un ConfigMap	<ol style="list-style-type: none"> <li>1. Seleccione el nombre del ConfigMap que se va a actualizar y haga clic en <b>Update</b>.</li> <li>2. Modifique los datos secretos. Para obtener más información, consulte <a href="#">Tabla 12-1</a>.</li> <li>3. Haga clic en <b>OK</b>.</li> </ol>
Eliminación de un ConfigMap	Seleccione la configuración que desea eliminar y haga clic en <b>Delete</b> . Siga las instrucciones para eliminar el ConfigMap.

## 12.2 Uso de un ConfigMap

Después de crear un ConfigMap, se puede utilizar en tres escenarios de carga de trabajo: variables de entorno, parámetros de línea de comandos y volúmenes de datos.

- [Definición de variables de entorno de carga de trabajo](#)
- [Definición de parámetros de línea de comandos](#)
- [Conexión de un ConfigMap al volumen de datos de carga de trabajo](#)

En el siguiente ejemplo se muestra cómo utilizar un ConfigMap.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: cce-configmap
data:
  SPECIAL_LEVEL: Hello
  SPECIAL_TYPE: CCE
```

**AVISO**

- Cuando se utiliza un ConfigMap en una carga de trabajo, la carga de trabajo y el ConfigMap deben estar en el mismo clúster y espacio de nombres.
- Cuando se monta un ConfigMap como volumen de datos y se actualiza el ConfigMap, Kubernetes actualiza los datos en el volumen de datos al mismo tiempo.  
 Para un volumen de datos de ConfigMap montado en modo **subPath**, Kubernetes no puede actualizar automáticamente los datos en el volumen de datos cuando se actualiza el ConfigMap.
- Cuando se utiliza un ConfigMap como variable de entorno, los datos no se actualizan automáticamente cuando se actualiza el ConfigMap. Para actualizar los datos, debe reiniciar el pod.

## Definición de variables de entorno de carga de trabajo

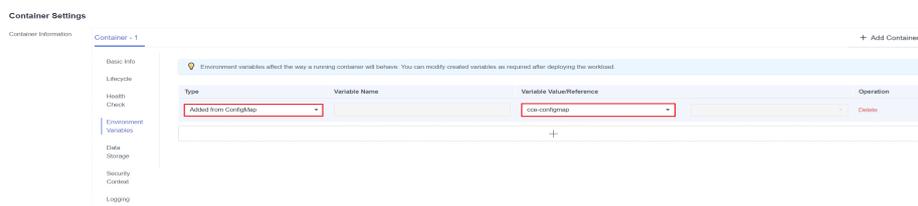
### Con la consola

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** En el panel de navegación, elija **Workloads**. A continuación, haga clic en **Create Workload**.

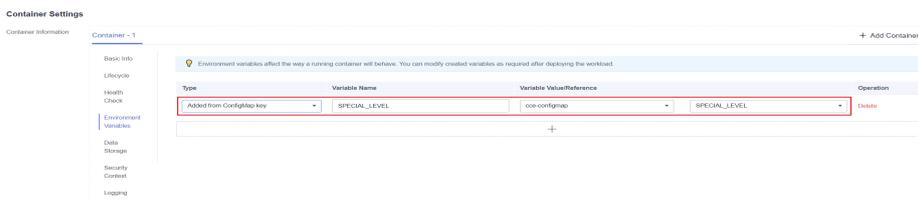
Cuando cree una carga de trabajo, haga clic en **Environment Variables** en el área **Container Settings** y haga clic en **+**.

- **Added from ConfigMap:** Seleccione un ConfigMap para importar todas sus claves como variables de entorno.



- **Added from ConfigMap key:** Importa una clave en un ConfigMap como el valor de una variable de entorno.
  - **Variable Name:** nombre de una variable de entorno en la carga de trabajo. El nombre se puede personalizar y se establece en el nombre de clave seleccionado en el ConfigMap de forma predeterminada.
  - **Variable Value/Reference:** Seleccione un ConfigMap y la clave que se va a importar. El valor correspondiente se importa como una variable de entorno de carga de trabajo.

Por ejemplo, después de importar el valor **Hello** de **SPECIAL\_LEVEL** en el ConfigMap **cc-c-configmap** como valor de la variable de entorno de carga de trabajo **SPECIAL\_LEVEL**, existe una variable de entorno llamada **SPECIAL\_LEVEL** con su valor **Hello** en el contenedor.



**Paso 3** Configure otros parámetros de carga de trabajo y haga clic en **Create Workload**.

Una vez que la carga de trabajo se ejecute correctamente, **inicie sesión en contenedor** y ejecute la siguiente instrucción para comprobar si ConfigMap se ha establecido como una variable de entorno de la carga de trabajo:

```
printenv SPECIAL_LEVEL
```

El resultado del ejemplo es el siguiente:

```
Hello
```

----Fin

**Con kubectl****Paso 1** Utilice kubectl para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).**Paso 2** Cree un archivo llamado **nginx-configmap.yaml** y edítelo.**vi nginx-configmap.yaml**

Contenido del archivo YAML:

- **Added from a ConfigMap:** Para agregar todos los datos de un ConfigMap a las variables de entorno, utilice el parámetro **envFrom**. Las claves en el ConfigMap se convertirán en nombres de variables de entorno en la carga de trabajo.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-configmap
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx-configmap
  template:
    metadata:
      labels:
        app: nginx-configmap
    spec:
      containers:
        - name: container-1
          image: nginx:latest
          envFrom:
            # Use envFrom to specify a ConfigMap to
            # be referenced by environment variables.
            - configMapRef:
                name: cce-configmap
                # Name of the referenced ConfigMap.
          imagePullSecrets:
            - name: default-secret
```

- **Added from a ConfigMap key:** Al crear una carga de trabajo, puede usar un ConfigMap para establecer variables de entorno y usar el parámetro **valueFrom** para hacer referencia al par clave-valor en ConfigMap por separado.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-configmap
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx-configmap
  template:
    metadata:
      labels:
```

```

    app: nginx-configmap
    spec:
      containers:
        - name: container-1
          image: nginx:latest
          env:
            # Set the environment variable in
            the workload.
            - name: SPECIAL_LEVEL
              # Name of the environment variable in
              the workload.
              valueFrom:
                # Specify a ConfigMap to be
                referenced by the environment variable.
                configMapKeyRef:
                  name: cce-configmap
                  key: SPECIAL_LEVEL
            - name: SPECIAL_TYPE
              # Add multiple environment variables
              to import them at the same time.
              valueFrom:
                configMapKeyRef:
                  name: cce-configmap
                  key: SPECIAL_TYPE
          imagePullSecrets:
            - name: default-secret
    
```

**Paso 3** Cree una carga de trabajo.

**kubectl apply -f nginx-configmap.yaml**

**Paso 4** Vea la variable de entorno en el pod.

1. Ejecute el siguiente comando para ver el pod creado:

```
kubectl get pod | grep nginx-configmap
```

Producto esperado:

```
nginx-configmap-*** 1/1 Running 0 2m18s
```

2. Ejecute el siguiente comando para ver las variables de entorno en el pod:

```
kubectl exec nginx-configmap-*** -- printenv SPECIAL_LEVEL SPECIAL_TYPE
```

Producto esperado:

```
Hello
CCE
```

El ConfigMap se ha establecido como variables de entorno de la carga de trabajo.

----Fin

## Definición de parámetros de línea de comandos

Puede utilizar un ConfigMap como variable de entorno para establecer comandos o valores de parámetros para un contenedor mediante la sintaxis de sustitución de variable de entorno \$ (VAR\_NAME).

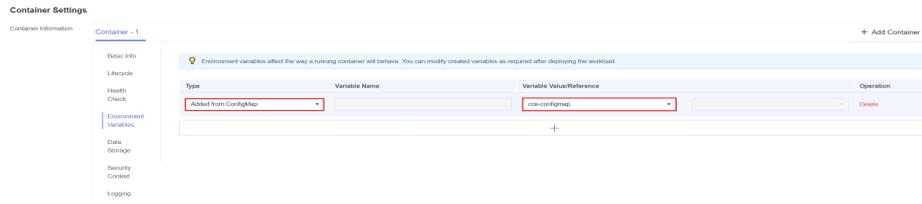
### Con la consola

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** En el panel de navegación, elija **Workloads**. A continuación, haga clic en **Create Workload**.

Cuando cree una carga de trabajo, haga clic en **Environment Variables** en el área **Container Settings** y haga clic en **+**. En este ejemplo, seleccione **Added from ConfigMap**.

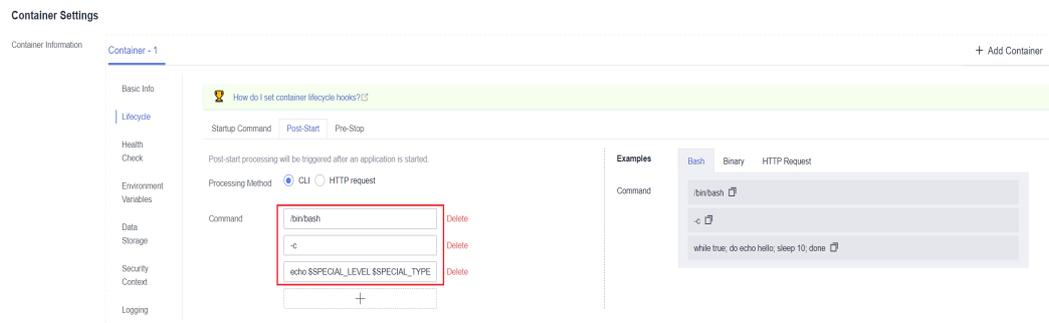
- **Added from ConfigMap**: Seleccione un ConfigMap para importar todas sus claves como variables de entorno.



**Paso 3** Haga clic en **Lifecycle** en el área **Container Settings**, haga clic en la ficha **Post-Start** de la derecha y establezca los siguientes parámetros:

- **Processing Method: CLI**
- **Command:** Ingrese las tres líneas de comando siguientes. *SPECIAL\_LEVEL* y *SPECIAL\_TYPE* son los nombres de variables de entorno de la carga de trabajo, es decir, los nombres clave del ConfigMap **cce-ConfigMap**.

```
/bin/bash
-c
echo $SPECIAL_LEVEL $SPECIAL_TYPE > /usr/share/nginx/html/index.html
```



**Paso 4** Defina otros parámetros de carga de trabajo y haga clic en **Create Workload**.

Una vez que la carga de trabajo se ejecute correctamente, **inicie sesión en contenedor** y ejecute la siguiente instrucción para comprobar si ConfigMap se ha establecido como una variable de entorno de la carga de trabajo:

```
cat /usr/share/nginx/html/index.html
```

El resultado del ejemplo es el siguiente:

```
Hello CCE
```

**----Fin**

**Con kubectl**

**Paso 1** Utilice kubectl para conectarse al clúster. Para obtener más información, véase **Conexión a un clúster con kubectl**.

**Paso 2** Cree un archivo llamado **nginx-configmap.yaml** y edítelo.

**vi nginx-configmap.yaml**

Como se muestra en el siguiente ejemplo, el ConfigMap **cce-ConfigMap** se importa a la carga de trabajo. *SPECIAL\_LEVEL* y *SPECIAL\_TYPE* son los nombres de variables de entorno de la carga de trabajo, es decir, los nombres clave del ConfigMap **cce-ConfigMap**.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-configmap
spec:
```

```

replicas: 1
selector:
  matchLabels:
    app: nginx-configmap
template:
  metadata:
    labels:
      app: nginx-configmap
  spec:
    containers:
      - name: container-1
        image: nginx:latest
        lifecycle:
          postStart:
            exec:
              command: [ "/bin/sh", "-c", "echo $SPECIAL_LEVEL $SPECIAL_TYPE
> /usr/share/nginx/html/index.html" ]
          envFrom:
            # Use envFrom to specify a ConfigMap to be
            # referenced by environment variables.
            - configMapRef:
                name: cce-configmap          # Name of the referenced ConfigMap.
        imagePullSecrets:
          - name: default-secret
    
```

**Paso 3** Cree una carga de trabajo.

**kubectl apply -f nginx-configmap.yaml**

**Paso 4** Una vez que la carga de trabajo se ejecuta correctamente, se introduce el siguiente contenido en el archivo `/usr/share/nginx/html/index.html`:

1. Ejecute el siguiente comando para ver el pod creado:

```
kubectl get pod | grep nginx-configmap
```

Producto esperado:

```
nginx-configmap-*** 1/1 Running 0 2m18s
```

2. Ejecute el siguiente comando para ver las variables de entorno en el pod:

```
kubectl exec nginx-configmap-*** -- cat /usr/share/nginx/html/index.html
```

Producto esperado:

```
Hello CCE
```

----Fin

## Conexión de un ConfigMap al volumen de datos de carga de trabajo

Los datos almacenados en un ConfigMap pueden ser referenciados en un volumen de tipo ConfigMap. Puede montar dicho volumen en una ruta de acceso de contenedor especificada. La plataforma admite la separación de códigos de carga de trabajo y archivos de configuración. Los volúmenes de ConfigMap se utilizan para almacenar parámetros de configuración de la carga de trabajo. Antes de eso, debe crear ConfigMaps por adelantado. Para obtener más información, véase [Creación de un ConfigMap](#).

### Con la consola

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** En el panel de navegación, elija **Workloads**. A continuación, haga clic en **Create Workload**.

Cuando cree una carga de trabajo, haga clic en **Data Storage** en el área **Container Settings**. Haga clic en **Add Volume** y seleccione **ConfigMap** en la lista desplegable.

**Paso 3** Establezca el tipo de volumen local en **ConfigMap** y establezca los parámetros para agregar un volumen local, como se muestra en [Tabla 12-4](#).

**Tabla 12-4** Montaje de un volumen de ConfigMap

Parámetro	Descripción
ConfigMap	<p>Seleccione el ConfigMap deseado.</p> <p>Un ConfigMap debe ser creado por adelantado. Para obtener más información, véase <a href="#">Creación de un ConfigMap</a>.</p>
Add Container Path	<p>Configure los parámetros siguientes:</p> <ol style="list-style-type: none"> <li> <p><b>Mount Path:</b> Introduzca una ruta de acceso del contenedor. Por ejemplo, <code>/tmp</code>.</p> <p>Este parámetro indica la ruta de contenedor en la que se montará un volumen de datos. No monte el volumen en un directorio del sistema como <code>/</code> o <code>/var/run</code>; esta acción puede causar errores de contenedor. Se recomienda montar el volumen en un directorio vacío. Si el directorio no está vacío, asegúrese de que no hay archivos que afecten al inicio de contenedor. De lo contrario, se reemplazarán los archivos, lo que provocará errores de inicio de contenedor o errores de creación de carga de trabajo.</p> <p><b>AVISO</b></p> <p>Cuando el contenedor está montado en un directorio de alto riesgo, se recomienda utilizar una cuenta con los permisos mínimos para iniciar el contenedor; de lo contrario, los archivos de alto riesgo en la máquina host podrían estar dañados.</p> </li> <li> <p><b>Subpath:</b> Introduzca una ruta secundaria, por ejemplo, <code>tmp</code>.</p> <ul style="list-style-type: none"> <li>Se utiliza una ruta secundaria para montar un volumen local de modo que se utilice el mismo volumen de datos en un solo pod. Si este parámetro se deja en blanco, la ruta raíz se utiliza de forma predeterminada.</li> <li>La ruta secundaria puede ser la clave y el valor de un ConfigMap o secreto. Si la ruta secundaria es un par clave-valor que no existe, la importación de datos no tiene efecto.</li> <li>Los datos importados especificando una ruta secundaria no se actualizarán junto con las actualizaciones de ConfigMap/secret.</li> </ul> </li> <li> <p>Establezca el permiso en <b>Read-only</b>. Los volúmenes de datos de la ruta son de solo lectura.</p> </li> </ol> <p>Puede hacer clic en  para agregar varias rutas y subrutas.</p>

----Fin

**Con kubectl**

**Paso 1** Utilice kubectl para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).

**Paso 2** Cree un archivo llamado `nginx-configmap.yaml` y edítelo.

**vi nginx-configmap.yaml**

Como se muestra en el siguiente ejemplo, después de montar el volumen de ConfigMap, se genera un archivo de configuración con la clave como el nombre y el valor como el contenido del archivo en el directorio `/etc/config` del contenedor.

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-configmap
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx-configmap
  template:
    metadata:
      labels:
        app: nginx-configmap
    spec:
      containers:
      - name: container-1
        image: nginx:latest
        volumeMounts:
        - name: config-volume
          mountPath: /etc/config          # Mount to the /etc/config directory.
          readOnly: true
      volumes:
      - name: config-volume
        configMap:
          name: cce-configmap          # Name of the referenced ConfigMap.
    
```

**Paso 3** Cree una carga de trabajo.

**kubectl apply -f nginx-configmap.yaml**

**Paso 4** Una vez que la carga de trabajo se ejecuta correctamente, los archivos **SPECIAL\_LEVEL** y **SPECIAL\_TYPE** se generan en el directorio **/etc/config**. El contenido de los archivos son **Hello** y **CCE** respectivamente.

1. Ejecute el siguiente comando para ver el pod creado:

```
kubectl get pod | grep nginx-configmap
```

Producto esperado:

```
nginx-configmap-*** 1/1 Running 0 2m18s
```

2. Ejecute el siguiente comando para ver el archivo **SPECIAL\_LEVEL** o **SPECIAL\_TYPE** en el pod:

```
kubectl exec nginx-configmap-*** -- /etc/config/SPECIAL_LEVEL
```

Producto esperado:

```
Hello
```

----Fin

## 12.3 Creación de un secreto

### Escenario

Un secreto es un tipo de recurso que contiene datos confidenciales, como la autenticación y la información clave. Su contenido está definido por el usuario. Después de crear secretos, puede usarlos como archivos o variables de entorno en una carga de trabajo contenedORIZADA.

### Restricciones

Los secretos no se pueden usar en **Pods estáticos**.

## Procedimiento

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** Elija **ConfigMaps and Secrets** en el panel de navegación, haga clic en la ficha **Secrets** y haga clic en **Create Secret** en la esquina superior derecha.

**Paso 3** Establezca los parámetros.

**Tabla 12-5** Parámetros para crear un secreto

Parámetro	Descripción
Name	Nombre del secreto que crea, que debe ser único.
Namespace	Espacio de nombres al que pertenece el secreto. Si no especifica este parámetro, el valor <b>default</b> se utiliza de forma predeterminada.
Description	Descripción de un secreto.
Type	Tipo del secreto que crea. <ul style="list-style-type: none"> <li>● Opaque: secreto común.</li> <li>● <code>kubernetes.io/dockerconfigjson</code>: un secreto que almacena la información de autenticación necesaria para extraer imágenes de un repositorio privado.</li> <li>● <code>kubernetes.io/tls</code>: secreto de TLS de Kubernetes, que se utiliza para almacenar el certificado requerido por los servicios de balanceo de carga de capa 7.</li> <li>● <b>IngressTLS</b>: secreto de TLS proporcionado por CCE para almacenar el certificado requerido por los servicios de balanceo de carga de capa 7.</li> <li>● Otro: otro tipo de secreto, que se especifica manualmente.</li> </ul>
Secret Data	Los datos de secretos de la carga de trabajo se pueden usar en contenedores. <ul style="list-style-type: none"> <li>● Si <b>Secret Type</b> es <b>Opaque</b>, haga clic en <b>+</b>. En el cuadro de diálogo que se muestra, introduzca un par clave-valor y seleccione <b>Auto Base64 Encoding</b>.</li> <li>● Si <b>Secret Type</b> es <code>kubernetes.io/dockerconfigjson</code>, introduzca la cuenta y la contraseña del repositorio de imágenes privadas.</li> <li>● Si <b>Secret Type</b> es <code>kubernetes.io/tls</code> o <b>IngressTLS</b>, suba el archivo de certificado y el archivo de clave privada.</li> </ul> <p><b>NOTA</b></p> <ul style="list-style-type: none"> <li>– Un certificado es una credencial autofirmada o firmada por CA utilizada para la autenticación de identidad.</li> <li>– Una solicitud de certificado es una solicitud de firma con una clave privada.</li> </ul>
Secret Label	Etiqueta del secreto. Ingrese un par clave-valor y haga clic en <b>Add</b> .

**Paso 4** Una vez completada la configuración, haga clic en **OK**.

El nuevo secreto se muestra en la lista de claves.

----Fin

## Ejemplo de configuración del archivo de recursos secreto

En esta sección se describen ejemplos de configuración de archivos de descripción de recursos secretos.

- Tipo de Opaque

El archivo **secret.yaml** se define como se muestra a continuación. El campo **data** se rellena como un par clave-valor, y el campo **value** debe codificarse usando Base64. Para obtener más información sobre el método de codificación de Base64, consulte [Codificación Base64](#).

```
apiVersion: v1
kind: Secret
metadata:
  name: mysecret          #Secret name
  namespace: default     #Namespace. The default value is default.
data:
  <your_key>: <your_value> # Enter a key-value pair. The value must be
  encoded using Base64.
type: Opaque
```

- Tipo de kubernetes.io/dockerconfigjson

El archivo **secret.yaml** se define como se muestra a continuación. El valor de **.dockerconfigjson** debe codificarse usando Base64. Para obtener más información, véase [Codificación Base64](#).

```
apiVersion: v1
kind: Secret
metadata:
  name: mysecret          #Secret name
  namespace: default     #Namespace. The default value is default.
data:
  .dockerconfigjson: eyJh***** # Content encoded using Base64.
type: kubernetes.io/dockerconfigjson
```

Para obtener el contenido **.dockerconfigjson**, realice los siguientes pasos:

- Obtenga la siguiente información de inicio de sesión del repositorio de imágenes.
  - Dirección del repositorio de imágenes: La sección usa *address* como ejemplo. Sustitúyalo por la dirección real.
  - Nombre de usuario: La sección usa *username* como ejemplo. Sustitúyalo con el nombre de usuario real.
  - Contraseña: La sección usa *password* como ejemplo. Reemplácelo con la contraseña real.
- Utilice Base64 para codificar el par clave-valor *username:password* y rellenar el contenido codificado en **3**.

```
echo -n "username:password" | base64
```

Salida del comando:

```
dXN1cm5hbWU6cGFzc3dvcnQ=
```

- Utilice Base64 para codificar el siguiente contenido de JSON:

```
echo -n '{"auths":{"address":
{"username":"username","password":"password","auth":"dXN1cm5hbWU6cGFzc3dvcnQ="}}}' | base64
```

Salida del comando:

```
eyJhdXRocyI6eyJhZGRyZXNzIjpb7InVzZXJlIjoidXNlcm5hbWUiLCJwYXNzd29yZCI6I  
nBhc3N3b3JkIiwiaXV0aCI6ImRYTmxjbTVoYldVNmNHRnpjM2R2Y21RPSJ9fX0=
```

El contenido codificado es el contenido **.dockerconfigjson**.

- Tipo de kubernetes.io/tls

El valor de **tls.crt** y **tls.key** debe codificarse mediante Base64. Para obtener más información, véase [Codificación Base64](#).

```
kind: Secret
apiVersion: v1
metadata:
  name: mysecret          #Secret name
  namespace: default     #Namespace. The default value is default.
data:
  tls.crt: LS0tLS1CRU*****FURSOtLS0t # Certificate content, which must be
  encoded using Base64.
  tls.key: LS0tLS1CRU*****VZLS0tLS0= # Private key content, which must be
  encoded using Base64.
type: kubernetes.io/tls
```

- Tipo IngressTLS

El valor de **tls.crt** y **tls.key** debe codificarse mediante Base64. Para obtener más información, véase [Codificación Base64](#).

```
kind: Secret
apiVersion: v1
metadata:
  name: mysecret          #Secret name
  namespace: default     #Namespace. The default value is default.
data:
  tls.crt: LS0tLS1CRU*****FURSOtLS0t # Certificate content, which must be
  encoded using Base64.
  tls.key: LS0tLS1CRU*****VZLS0tLS0= # Private key content, which must be
  encoded using Base64.
type: IngressTLS
```

## Creación de un secreto con kubectl

**Paso 1** De acuerdo con [Conexión a un clúster con kubectl](#), configure el comando **kubectl** para conectar un ECS al clúster.

**Paso 2** Cree y edite el archivo **cce-secret.yaml** codificado en Base64.

```
# echo -n "content to be encoded" | base64
*****
```

**vi cce-secret.yaml**

El siguiente archivo YAML utiliza el tipo Opaque como ejemplo. Para obtener más información sobre otros tipos, consulte [Ejemplo de configuración del archivo de recursos secreto](#).

```
apiVersion: v1
kind: Secret
metadata:
  name: mysecret
type: Opaque
data:
  <your_key>: <your_value> # Enter a key-value pair. The value must be encoded
  using Base64.
```

**Paso 3** Cree un secreto.

**kubectl create -f cce-secret.yaml**

Puede consultar el secreto después de la creación.

**kubectl get secret -n default**

----Fin

## Operaciones relacionadas

Después de crear un secreto, puede actualizarlo o eliminarlo como se describe en [Tabla 12-6](#).

### NOTA

La lista secreta contiene recursos secretos del sistema que solo se pueden consultar. Los recursos secretos del sistema no se pueden actualizar ni eliminar.

**Tabla 12-6** Operaciones relacionadas

Operación	Descripción
Edición de un archivo YAML	Haga clic en <b>Edit YAML</b> en la fila donde reside el secreto de destino para editar su archivo YAML.
Actualización de un secreto	<ol style="list-style-type: none"> <li>1. Seleccione el nombre del secreto que desea actualizar y haga clic en <b>Update</b>.</li> <li>2. Modifique los datos secretos. Para obtener más información, consulte <a href="#">Tabla 12-5</a>.</li> <li>3. Haga clic en <b>OK</b>.</li> </ol>
Eliminación de un secreto	<p>Seleccione el secreto que desea eliminar y haga clic en <b>Delete</b>.</p> <p>Siga las instrucciones para eliminar el secreto.</p>
Eliminación de secretos por lotes	<ol style="list-style-type: none"> <li>1. Seleccione los secretos que desea eliminar.</li> <li>2. Haga clic en <b>Delete</b> encima de la lista secreta.</li> <li>3. Siga las instrucciones para eliminar los secretos.</li> </ol>

## Codificación Base64

Para codificar una cadena a Base64, ejecute el comando **echo -n "content to be encoded" | base64**. A continuación se presenta un ejemplo:

```
root@ubuntu:~# echo -n "content to be encoded" | base64
*****
```

## 12.4 Uso de un secreto

Después de crear los secretos, pueden montarse como volúmenes de datos o exponerse como variables de entorno para ser utilizadas por un contenedor en un pod.

**AVISO**

No realice ninguna operación sobre los siguientes secretos. Para obtener más información, véase [Secretos de clúster](#).

- No opere secretos bajo el kube-system.
- No utilice default-secret y paas.elb en ninguno de los espacios de nombres. El default-secret se utiliza para extraer la imagen privada de SWR, y el paas.elb se utiliza para conectar el servicio en el espacio de nombres al servicio de ELB.

- **Definición de variables de entorno de una carga de trabajo**
- **Configuración del volumen de datos de una carga de trabajo**

En el siguiente ejemplo se muestra cómo utilizar un secreto.

```
apiVersion: v1
kind: Secret
metadata:
  name: mysecret
type: Opaque
data:
  username: ***** #The value must be Base64-encoded.
  password: ***** #The value must be encoded using Base64.
```

**AVISO**

- Cuando se usa un secreto en un pod, el pod y el secreto deben estar en el mismo clúster y espacio de nombres.
- Cuando se actualiza un secreto, Kubernetes actualiza los datos en el volumen de datos al mismo tiempo.  
 Sin embargo, cuando se actualiza un volumen de datos secreto montado en modo **subPath**, Kubernetes no puede actualizar automáticamente los datos en el volumen de datos.

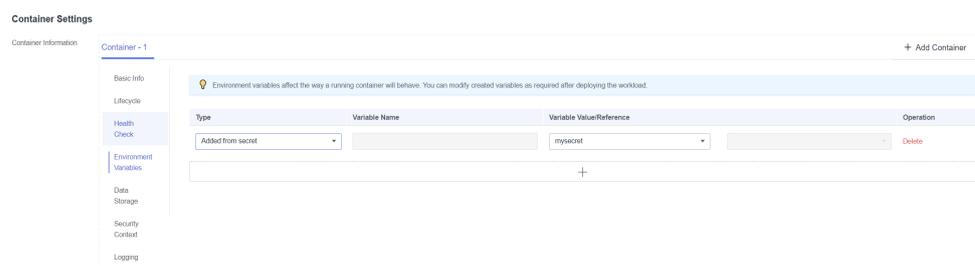
## Definición de variables de entorno de una carga de trabajo

### Con la consola

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

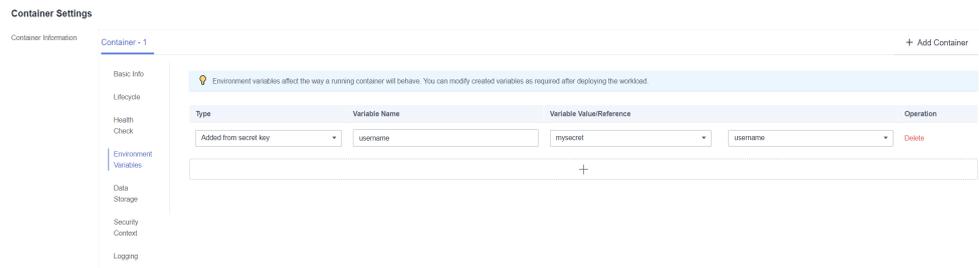
**Paso 2** En el panel de navegación, elija **Workloads**. A continuación, haga clic en **Create Workload**. Cuando cree una carga de trabajo, haga clic en **Environment Variables** en el área **Container Settings** y haga clic en **+**.

- **Added from secret:** Seleccione un secreto e importe todas las claves del secreto como variables de entorno.



- **Added from secret key:** Importa el valor de una clave en un secreto como el valor de una variable de entorno.
  - **Variable Name:** nombre de una variable de entorno en la carga de trabajo. El nombre se puede personalizar y se establece en el nombre de clave seleccionado en secreto de forma predeterminada.
  - **Variable Value/Reference:** Seleccione un secreto y la clave que se va a importar. El valor correspondiente se importa como una variable de entorno de carga de trabajo.

Por ejemplo, después de importar el valor de **username** en el secreto **mysecret** como el valor de la variable de entorno de carga de trabajo **username**, existe una variable de entorno denominada **username** en el contenedor.



**Paso 3** Defina otros parámetros de carga de trabajo y haga clic en **Create Workload**.

Una vez que la carga de trabajo se ejecute correctamente, **inicie sesión en contenedor** y ejecute la siguiente instrucción para comprobar si el secreto se ha establecido como una variable de entorno de la carga de trabajo:

```
printenv username
```

Si el resultado es el mismo que el contenido del secreto, el secreto se ha establecido como una variable de entorno de la carga de trabajo.

----Fin

**Con kubectl**

**Paso 1** Utilice kubectl para conectarse al clúster. Para obtener más información, véase **Conexión a un clúster con kubectl**.

**Paso 2** Cree un archivo llamado **nginx-secret.yaml** y edítelo.

**vi nginx-secret.yaml**

Contenido del archivo YAML:

- **Added from a secret:** Para agregar todos los datos de un secreto a las variables de entorno, utilice el parámetro **envFrom**. Las claves secretas se convertirán en nombres de variables de entorno en una carga de trabajo.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-secret
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx-secret
  template:
    metadata:
```

```

labels:
  app: nginx-secret
spec:
  containers:
  - name: container-1
    image: nginx:latest
    envFrom:
      # Use envFrom to specify a secret to be
      # referenced by environment variables.
      - secretRef:
          name: mysecret
          # Name of the referenced secret.
    imagePullSecrets:
    - name: default-secret
    
```

- **Added from a secret key:** Al crear una carga de trabajo, puede usar un secreto para establecer variables de entorno y usar el parámetro **valueFrom** para hacer referencia al par clave-valor en el secreto por separado.

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-secret
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx-secret
  template:
    metadata:
      labels:
        app: nginx-secret
    spec:
      containers:
      - name: container-1
        image: nginx:latest
        env:
          # Set the environment variable in
          # the workload.
          - name: SECRET_USERNAME
            # Name of the environment variable
            # in the workload.
            valueFrom:
              # Use valueFrom to specify a secret
              # to be referenced by environment variables.
              secretKeyRef:
                name: mysecret
                # Name of the referenced secret.
                key: username
                # Key in the referenced secret.
          - name: SECRET_PASSWORD
            # Add multiple environment
            # variables to import them at the same time.
            valueFrom:
              secretKeyRef:
                name: mysecret
                key: password
        imagePullSecrets:
        - name: default-secret
    
```

**Paso 3** Cree una carga de trabajo.

```
kubectl apply -f nginx-secret.yaml
```

**Paso 4** Vea las variables de entorno en el pod.

1. Ejecute el siguiente comando para ver el pod creado:

```
kubectl get pod | grep nginx-secret
```

Producto esperado:

```
nginx-secret-*** 1/1 Running 0 2m18s
```

2. Ejecute el siguiente comando para ver las variables de entorno en el pod:

```
kubectl exec nginx-secret-*** -- printenv SPECIAL_USERNAME SPECIAL_PASSWORD
```

Si el resultado es el mismo que el contenido del secreto, el secreto se ha establecido como una variable de entorno de la carga de trabajo.

----Fin

## Configuración del volumen de datos de una carga de trabajo

Puede montar un secreto como un volumen en la ruta especificada de acceso de contenedor. El contenido de un secreto está definido por el usuario. Antes de eso, necesita crear un secreto. Para obtener más información, véase [Creación de un secreto](#).

### Con la consola

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** En el panel de navegación de la izquierda, haga clic en **Workloads**. En el panel derecho, haga clic en la ficha **Deployments**. Haga clic en **Create Workload** en la esquina superior derecha.

Cuando cree una carga de trabajo, haga clic en **Data Storage** en el área **Container Settings**. Haga clic en **Add Volume** y seleccione **Secret** en la lista desplegable.

**Paso 3** Establezca el tipo de volumen local en **Secret** y establezca los parámetros para agregar un volumen local, como se muestra en [Tabla 12-7](#).

**Tabla 12-7** Montar un volumen de secreto

Parámetro	Descripción
Secret	Seleccione el secreto deseado. Un secreto debe ser creado por adelantado. Para obtener más información, véase <a href="#">Creación de un secreto</a> .

Parámetro	Descripción
Add Container Path	<p>Configure los parámetros siguientes:</p> <ol style="list-style-type: none"> <li><b>Mount Path:</b> Introduzca una ruta de acceso del contenedor. Por ejemplo, <code>/tmp</code>.                      Este parámetro indica la ruta de contenedor en la que se montará un volumen de datos. No monte el volumen en un directorio del sistema como <code>/</code> o <code>/var/run</code>; esta acción puede causar errores de contenedor. Se recomienda montar el volumen en un directorio vacío. Si el directorio no está vacío, asegúrese de que no hay archivos que afecten al inicio de contenedor. De lo contrario, se reemplazarán los archivos, lo que provocará errores de inicio de contenedor o errores de creación de carga de trabajo.  <b>AVISO</b>                      Cuando el contenedor está montado en un directorio de alto riesgo, se recomienda utilizar una cuenta con los permisos mínimos para iniciar el contenedor; de lo contrario, los archivos de alto riesgo en la máquina host podrían estar dañados.</li> <li><b>Subpath:</b> Introduzca una ruta secundaria, por ejemplo, <code>tmp</code>.                     <ul style="list-style-type: none"> <li>Se utiliza una ruta secundaria para montar un volumen local de modo que se utilice el mismo volumen de datos en un solo pod. Si este parámetro se deja en blanco, la ruta raíz se utiliza de forma predeterminada.</li> <li>La ruta secundaria puede ser la clave y el valor de un ConfigMap o secreto. Si la ruta secundaria es un par clave-valor que no existe, la importación de datos no tiene efecto.</li> <li>Los datos importados especificando una ruta secundaria no se actualizarán junto con las actualizaciones de ConfigMap/secret.</li> </ul> </li> <li>Establezca el permiso en <b>Read-only</b>. Los volúmenes de datos de la ruta son de solo lectura.</li> </ol> <p>Puede hacer clic en  para agregar varias rutas y subrutas.</p>

---Fin

### Con kubectl

**Paso 1** Utilice kubectl para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).

**Paso 2** Cree un archivo llamado `nginx-secret.yaml` y edítelo.

#### vi nginx-secret.yaml

En el siguiente ejemplo, el nombre de usuario y la contraseña del secreto `mysecret` se guardan en el directorio `/etc/fo` como archivos.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-secret
spec:
  replicas: 1
  selector:
```

```

matchLabels:
  app: nginx-secret
template:
  metadata:
    labels:
      app: nginx-secret
  spec:
    containers:
      - name: container-1
        image: nginx:latest
        volumeMounts:
          - name: foo
            mountPath: /etc/foo          # Mount to the /etc/foo directory.
            readOnly: true
    volumes:
      - name: foo
        secret:
          secretName: mysecret          # Name of the referenced secret.
    
```

También puede utilizar el campo **items** para controlar la ruta de asignación de claves secretas. Por ejemplo, almacene el nombre de usuario en el directorio **/etc/foo/my-group/my-username** en el contenedor.

 **NOTA**

- Si utiliza el campo **items** para especificar la ruta de asignación de las claves secretas, las claves no especificadas no se crearán como archivos. Por ejemplo, si no se especifica la clave **password** del siguiente ejemplo, no se creará el archivo.
- Si desea utilizar todas las claves en un secreto, debe enumerar todas las claves en el campo **items**.
- Todas las claves enumeradas en el campo **items** deben existir en el secreto correspondiente. De lo contrario, el volumen no se crea.

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-secret
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx-secret
  template:
    metadata:
      labels:
        app: nginx-secret
    spec:
      containers:
        - name: container-1
          image: nginx:latest
          volumeMounts:
            - name: foo
              mountPath: /etc/foo          # Mount to the /etc/foo directory.
              readOnly: true
      volumes:
        - name: foo
          secret:
            secretName: mysecret          # Name of the referenced secret.
            items:
              - key: username              # Name of the referenced key.
                path: my-group/my-username # Mapping path of the secret key
    
```

**Paso 3** Cree una carga de trabajo.

**kubectl apply -f nginx-secret.yaml**

**Paso 4** Una vez que la carga de trabajo se ejecuta correctamente, los archivos **username** y **password** se generan en el directorio **/etc/foo**.

1. Ejecute el siguiente comando para ver el pod creado:

```
kubectl get pod | grep nginx-secret
```

Producto esperado:

```
nginx-secret-*** 1/1 Running 0 2m18s
```

2. Ejecute el siguiente comando para ver el archivo **username** o **password** en el pod:

```
kubectl exec nginx-secret-*** -- /etc/foo/username
```

El resultado esperado es el mismo que el contenido en secreto.

----Fin

## 12.5 Secretos de clúster

De forma predeterminada, CCE crea los siguientes secretos en cada espacio de nombres:

- default-secret
- paas.elb
- default-token-xxxxx (xxxxx is a random number.)

Las funciones de estos secretos se describen a continuación.

### default-secret

El tipo de **default-secret** es **kubernetes.io/dockerconfigjson**. Los datos son la credencial para iniciar sesión en el repositorio de imágenes de SWR y se utilizan para extraer imágenes de SWR. Si necesita extraer una imagen de SWR al crear una carga de trabajo en CCE, establezca **imagePullSecrets** en **default-secret**.

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - image: nginx:alpine
    name: container-0
  resources:
    limits:
      cpu: 100m
      memory: 200Mi
    requests:
      cpu: 100m
      memory: 200Mi
  imagePullSecrets:
  - name: default-secret
```

Los datos de **default-secret** se actualizan periódicamente, y los datos actuales caducan después de un cierto período de tiempo. Puede ejecutar el comando **describe** para ver el tiempo de caducidad de default-secret.

### AVISO

Utilice default-secret directamente en lugar de copiar el contenido secreto para crear uno nuevo. La credencial en el secreto copiado caducará y la imagen no se puede extraer.

```
$ kubectl describe secret default-secret
Name:          default-secret
```

```
Namespace:    default
Labels:       secret-generated-by=cce
Annotations:  temporary-ak-sk-expires-at: 2021-11-26 20:55:31.380909 +0000 UTC

Type:    kubernetes.io/dockerconfigjson

Data
====
.dockerconfigjson:  347 bytes
```

## paas.elb

Los datos de **paas.elb** son los datos temporales de AK/SK, que se utilizan para crear balanceadores de carga ELB durante la creación de Service y entrada. Los datos de paas.elb se actualizan periódicamente y caducan después de un cierto período de tiempo.

En la práctica, no va a utilizar directamente paas.elb. Sin embargo, no lo elimine. De lo contrario, los balanceadores de carga de ELB no se crearán.

## default-token-xxxxx

De forma predeterminada, Kubernetes crea una cuenta de servicio llamada **default** para cada espacio de nombres. **default-token-xxxxx** es la clave de la cuenta de servicio, y **xxxxx** es un número aleatorio.

```
$ kubectl get sa
NAME          SECRETS  AGE
default      1        30d
$ kubectl describe sa default
Name:         default
Namespace:    default
Labels:       <none>
Annotations:  <none>
Image pull secrets: <none>
Mountable secrets:  default-token-vssmw
Tokens:       default-token-vssmw
Events:       <none>
```

# 13 Auto Scaling

---

## 13.1 Descripción general

El ajuste automático es un servicio que ajusta los recursos de servicio de forma automática y económica en función de los requisitos de servicio y las políticas configuradas.

### Contexto

Cada vez se desarrollan más aplicaciones basadas en Kubernetes. Cada vez es más importante escalar rápidamente las aplicaciones en Kubernetes para hacer frente a los picos de servicio y escalar las aplicaciones durante las horas no pico para ahorrar recursos y reducir costos.

En un clúster de Kubernetes, el ajuste automático implica pods y nodos. Un pod es una instancia de aplicación. Cada pod contiene uno o más contenedores y se ejecuta en un nodo (VM o servidor de metal puro). Si un clúster no tiene suficientes nodos para ejecutar nuevos pods, debe agregar nodos al clúster para garantizar la ejecución del servicio.

En CCE, el ajuste automático se utiliza para servicios en línea, cómputo y entrenamiento a gran escala, GPU de aprendizaje profundo o entrenamiento e inferencia de GPU compartida, cambios de carga periódicos y muchos otros escenarios.

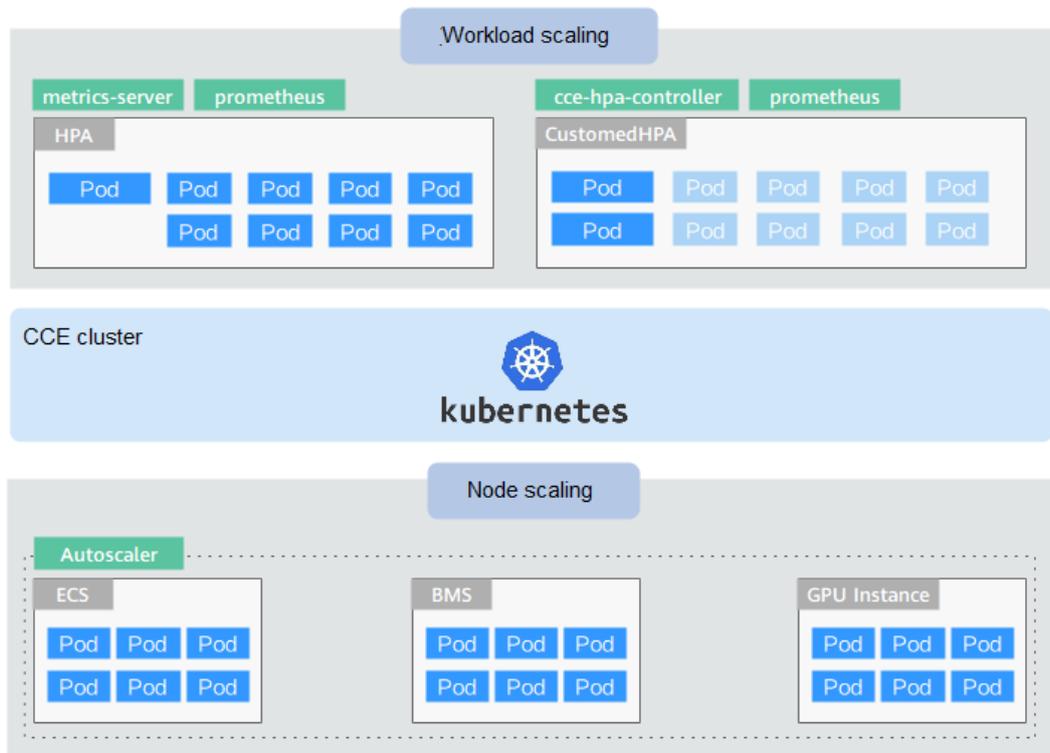
### Ajuste automático en CCE

**CCE admite el ajuste automático para las cargas de trabajo y los nodos.**

- **Workload scaling:** Ajuste automático en la capa de programación para cambiar la capacidad de programación de las cargas de trabajo. Por ejemplo, puede utilizar el HPA, un componente de ajuste en la capa de programación, para ajustar el número de réplicas de una aplicación. El ajuste del número de réplicas cambia la capacidad de programación ocupada por la carga de trabajo actual, lo que permite escalar en la capa de programación.
- **Node scaling:** Ajuste automático en la capa de recursos. Cuando los nodos de clúster planificados no pueden permitir la programación de la carga de trabajo, se proporcionan recursos de ECS para admitir la programación.

El ajuste de la carga de trabajo y el ajuste de nodos pueden funcionar por separado o en conjunto. Para obtener más información, véase [Uso de HPA y CA para el ajuste automático de cargas de trabajo y nodos](#).

## Componentes



Los componentes de ajuste de carga de trabajo se describen de la siguiente manera:

**Tabla 13-1** Componentes de ajuste de carga de trabajo

Tipo	Nombre del componente	Descripción de los componentes	Referencia
HPA	<a href="#">metrics-server</a>	Un componente integrado de Kubernetes, que permite el ajuste horizontal los pods. Agrega la ventana de tiempo de enfriamiento a nivel de aplicación y las funciones de umbral de ajuste basadas en el HPA.	<a href="#">Creación de una política de HPA para el escalado automático de cargas de trabajo</a>
CustomedHPA	<a href="#">cce-hpa-controller</a>	Una función de ajuste automático mejorada, utilizada para ajuste automático de Deployments basado en métricas (uso de CPU y uso de memoria) o en un intervalo periódico (un punto de tiempo específico cada día, cada semana, cada mes o cada año).	<a href="#">Creación de una política de CustomedHPA para el ajuste automático de cargas de trabajo</a>

Tipo	Nombre del componente	Descripción de los componentes	Referencia
	<a href="#">prometheus</a> <a href="#">kube-prometheus-stack</a>	Un marco de supervisión y alarma de sistema de código abierto, que recopila métricas públicas (uso de CPU y uso de memoria) de kubelet en el clúster de Kubernetes.	

Los componentes de ajuste de nodos se describen a continuación:

Tabla 13-2 Componentes de ajuste de nodos

Nombre del componente	Descripción de los componentes	Escenario de aplicación	Referencia
<a href="#">autoscaler</a>	Un componente de Kubernetes de código abierto para el ajuste horizontal de nodos, que está optimizado en términos de capacidades de programación y ajuste automático.	Servicios en línea, aprendizaje profundo e informática a gran escala con presupuestos de recursos limitados	<a href="#">Creación de una política de ajuste de nodos</a>

## 13.2 Scaling a Workload

### 13.2.1 Mecanismos de ajuste de la carga de trabajo

CCE admite políticas de HPA y de CustomedHPA para escalar cargas de trabajo. En la siguiente tabla se describen las diferencias entre estos dos tipos de políticas.

Tabla 13-3 Comparación entre las políticas de HPA y de CustomedHPA

Concepto	Política de HPA	Política de CustomedHPA
Implementación	Kubernetes Horizontal Pod Autoscaling	Capacidades de ajuste automático mejoradas
Reglas	Escala las Deployments según <b>metrics</b> (el uso de la CPU y de la memoria).	Escala las Deployments según <b>metrics</b> (el uso de la CPU y de la memoria) o en un intervalo de <b>periodic</b> (un punto de tiempo específico cada día, cada semana, cada mes o cada año).

Concepto	Política de HPA	Política de CustomedHPA
Mejoras	Agrega la ventana de tiempo de enfriamiento a nivel de aplicación y las funciones de umbral de ajuste basadas en el HPA de Kubernetes.	<p><b>Metric-based:</b></p> <ul style="list-style-type: none"> <li>● El ajuste se puede realizar basándose en el porcentaje del número actual de pods.</li> <li>● Se puede establecer la etapa del ajuste mínimo. El ajuste se puede realizar paso a paso.</li> <li>● Se pueden realizar diferentes operaciones de ajuste según los valores métricos reales.</li> </ul> <p><b>Periodic:</b></p> <p>Puede seleccionar un punto de tiempo específico todos los días, cada semana, cada mes o cada año o un período como el tiempo de activación.</p>

## Cómo funciona HPA

HPA es un controlador que controla el ajuste horizontal de pod. HPA comprueba periódicamente las métricas de pod, calcula el número de réplicas necesarias para cumplir con los valores de destino configurados para los recursos de HPA y, a continuación, ajusta el valor del campo **replicas** en el objeto de recurso de destino (como una Deployment).

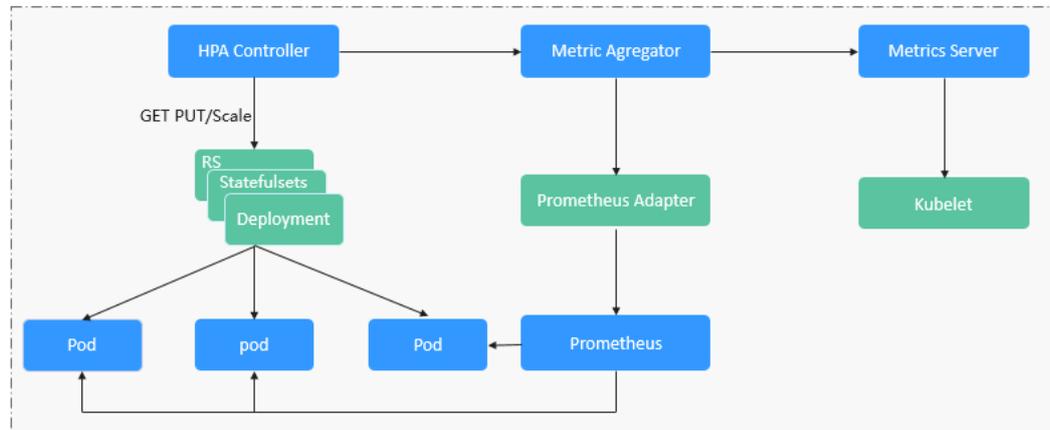
Un requisito previo para el ajuste automático es que se puedan recopilar los datos de ejecución de contenedor, como el número de nodos/pods de clúster y el uso de CPU y de memoria de contenedores. Kubernetes no proporciona tales capacidades de supervisión en sí. Puede utilizar extensiones para monitorear y recopilar sus datos. CCE integra **Prometheus** y **Metrics Server** para realizar estas capacidades:

- **Prometheus** es un marco de monitoreo de código abierto y alarmante que puede recopilar múltiples tipos de métricas. Prometheus ha sido una solución de monitoreo estándar de Kubernetes.
- **Metrics Server** es un agregador de datos de uso de recursos para todo el clúster. Metrics Server recopila métricas de la API de resumen expuesta por kubelet. Estas métricas se establecen para los recursos principales de Kubernetes, como pods, nodos, contenedores y Services. Metrics Server proporciona un conjunto de API estándar para que los sistemas externos recopilen estas métricas.

HPA puede trabajar con Metrics Server para implementar el ajuste automático basado en el uso de CPU y memoria. También puede trabajar con Prometheus para implementar ajuste automático basado en métricas de monitoreo personalizadas.

**Figura 13-1** muestra cómo funciona HPA.

**Figura 13-1** Proceso de trabajo de HPA



**Dos módulos principales de HPA:**

- Supervisión del origen de datos
 

La comunidad solo proporcionaba HPA basado en CPU y en memoria en la etapa inicial. Con la población de Kubernetes y Prometheus, los desarrolladores necesitan más métricas personalizadas o información de monitorización en la capa de acceso para sus propias aplicaciones, por ejemplo, el QPS del balanceador de carga y el número de usuarios en línea del sitio web. En respuesta, la comunidad define un conjunto de API de métricas estándar para proporcionar servicios externamente con estas API agregadas.

  - **metrics.k8s.io** proporciona métricas de monitorización relacionadas con la CPU y la memoria de pods y nodos.
  - **custom.metrics.k8s.io** proporciona métricas de supervisión personalizadas relacionadas con los objetos de Kubernetes.
  - **external.metrics.k8s.io** proporciona métricas que provienen de sistemas externos y son irrelevantes para cualquier métrica de recursos de Kubernetes.
- Algoritmos de toma de decisiones de ajuste
 

El controlador de HPA calcula la relación de ajuste en función de los valores métricos actuales y los valores métricos deseados utilizando la siguiente fórmula:

**desiredReplicas = ceil[currentReplicas x (currentMetricValue/desiredMetricValue)]**

Por ejemplo, si el valor métrico actual es 200m y el valor objetivo es 100m, el número deseado de pods se duplicará de acuerdo con la fórmula. En la práctica, los pods pueden agregarse o reducirse constantemente. Para garantizar la estabilidad, el controlador de HPA está optimizado desde los siguientes aspectos:

  - Intervalo de enfriamiento: En la versión 1.11 y versiones anteriores, Kubernetes introdujo los parámetros de inicio **horizontal-pod-autoscaler-downscale-stabilization-window** y **horizontal-pod-autoScaler-upscale-stabilization-window** para indicar los intervalos de enfriamiento después de una reducción y una expansión respectivamente, en los que no se realizará ninguna operación de ajuste. En versiones posteriores a la v1.14, la cola de programación se introduce para almacenar todas las sugerencias de toma de decisiones detectadas dentro de un periodo de tiempo. A continuación, el sistema toma decisiones basadas en todas las sugerencias de toma de decisiones válidas para minimizar los cambios del número deseado de réplicas para garantizar la estabilidad.
  - Tolerancia: Se puede considerar como una zona de búfer. Si se pueden tolerar cambios en el número de pods, el número de pods permanece sin cambios.

Utilice la fórmula:  $\text{ratio} = \text{currentMetricValue} / \text{desiredMetricValue}$

Cuando  $|\text{ratio} - 1.0| \leq \text{tolerancia}$ , no se realizará el ajuste.

Cuando  $|\text{ratio} - 1.0| > \text{tolerancia}$ , el valor deseado se calcula utilizando la fórmula mencionada anteriormente.

El valor predeterminado es 0.1 en la versión actual de la comunidad.

HPA realiza el ajuste basándose en los umbrales métricos. Las métricas comunes incluyen el uso de CPU y de memoria. También puede establecer métricas personalizadas, como el QPS y el número de conexiones, para activar el ajuste. Sin embargo, el ajuste basado en métricas trae la latencia de minutos generados durante las fases de recopilación, determinación y ajuste de datos. Tal latencia puede causar un alto uso de la CPU y una respuesta lenta. Para resolver este problema, CCE le permite configurar políticas programadas para escalar recursos regularmente para aplicaciones con los cambios periódicos.

## 13.2.2 Creación de una política de HPA para el escalado automático de cargas de trabajo

Horizontal Pod Autoscaling (HPA) en Kubernetes implementa el ajuste horizontal de pods. En una política de HPA de CCE, puede configurar diferentes ventanas de tiempo de enfriamiento y umbrales de ajuste para diferentes aplicaciones basadas en el HPA de Kubernetes.

### Requisitos previos

Para usar HPA, necesita instalar un complemento que proporcione API de métricas. Seleccione uno de los siguientes complementos según la versión del clúster y los requisitos reales.

- **metrics-server**: proporciona las métricas básicas de uso de recursos, como CPU de contenedor y uso de memoria. Es compatible con todas las versiones de clúster.
- **prometheus**: proporciona las métricas personalizadas además de las métricas básicas de recursos. Es necesario registrar Prometheus como el servicio que proporciona API de métricas. Para obtener más información, véase [Proporcionar métricas de recursos](#). Este complemento solo admite clústeres de v1.21 o anteriores.
- **kube-prometheus-stack**: proporciona las métricas personalizadas además de las métricas básicas de recursos. Es necesario registrar Prometheus como el servicio que proporciona API de métricas. Para obtener más información, véase [Proporcionar métricas de recursos](#). Este complemento solo admite clústeres de v1.23 o posterior.

### Notas y restricciones

- Las políticas de HPA solo se pueden crear para clústeres de v1.13 o posterior.
- Para los clústeres anteriores a v1.19.10, si se utiliza una política de HPA para escalar una carga de trabajo con volúmenes de EVS montados, los pods existentes no se pueden leer ni escribir cuando se programa un nuevo pod en otro nodo.

Para los clústeres de v1.19.10 y las versiones posteriores, si se utiliza una política de HPA para escalar una carga de trabajo con un volumen de EVS montado, no se puede iniciar un nuevo pod porque no se pueden conectar los discos de EVS.

### Procedimiento

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** En el panel de navegación, elija **Workload Scaling**. A continuación, haga clic en **Create HPA Policy** en la esquina superior derecha.

**Paso 3** Establezca los parámetros de política.

**Tabla 13-4** Parámetros de política de HPA

Parámetro	Descripción
Policy Name	Nombre de la política que se va a crear. Configure este parámetro según sea necesario.
Namespace	Espacio de nombres al que pertenece la carga de trabajo.
Associated Workload	Carga de trabajo con la que está asociada la política de HPA.
Pod Range	Número mínimo y máximo de pods. Cuando se activa una política, los pods de carga de trabajo se escalan dentro de este intervalo.
Cooldown Period	Intervalo entre una reducción y una expansión. La unidad es un minuto. <b>El intervalo no puede ser inferior a 1 minuto.</b> <b>Este parámetro solo se admite en los clústeres de v1.15 a v1.23.</b> Este parámetro indica el intervalo entre las operaciones de ajuste consecutivas. El período de tiempo de reutilización garantiza que se inicie una operación de ajuste solo cuando se haya completado la anterior y el sistema se esté ejecutando de manera estable.
Scaling Behavior	<b>Este parámetro solo se admite en los clústeres de v1.25 o posterior.</b> <ul style="list-style-type: none"> <li>● <b>Default:</b> Escala las cargas de trabajo utilizando el comportamiento predeterminado de Kubernetes. Para obtener más información, consulte <a href="#">Comportamiento predeterminado</a>.</li> <li>● <b>Custom:</b> Escala las cargas de trabajo mediante políticas personalizadas, como ventana de estabilización, pasos y prioridades. Los parámetros no especificados utilizan los valores recomendados por Kubernetes.                             <ul style="list-style-type: none"> <li>– <b>Disable scale-out/scale-in:</b> Seleccione si desea deshabilitar la expansión o reducción horizontal.</li> <li>– <b>Stabilization Window:</b> Período durante el cual CCE comprueba continuamente si las métricas utilizadas para escalar siguen fluctuando. CCE activa la escala si no se mantiene el estado deseado para toda la ventana. Esta ventana restringe el aleteo no deseado del recuento de pod debido a cambios de métrica.</li> <li>– <b>Step:</b> especifica el paso de ajuste. Puede establecer el número o porcentaje de pods que se ajustarán o reducirán dentro de un período especificado. Si hay varias políticas, puede seleccionar la política que maximiza o minimiza el número de pods.</li> </ul> </li> </ul>

Parámetro	Descripción
System Policy	<ul style="list-style-type: none"> <li>● <b>Metric:</b> Puede seleccionar <b>CPU usage</b> o <b>Memory usage</b>.</li> </ul> <p><b>NOTA</b>                      Uso = CPU o memorias utilizadas por pods/CPU o memorias solicitadas.</p> <ul style="list-style-type: none"> <li>● <b>Desired Value:</b> Introduzca el uso promedio de recursos deseado. Este parámetro indica el valor deseado de la métrica seleccionada. <math>\text{Número de pods a escalar (redondeado)} = (\text{Valor métrico actual} / \text{Valor deseado}) \times \text{Número de pods actuales}</math></li> </ul> <p><b>NOTA</b>                      Al calcular el número de pods que se agregarán o reducirán, la política de HPA utiliza el número máximo de pods en los últimos 5 minutos.</p> <ul style="list-style-type: none"> <li>● <b>Tolerance Range:</b> El escalado no se activa cuando el valor de la métrica está dentro del rango de tolerancia. El valor deseado debe estar dentro del rango de tolerancia. Si el valor de la métrica es mayor que el umbral de reducción y menor que el umbral de expansión, no se activa el ajuste. <b>Este parámetro solo se admite en los clústeres de v1.15 o posterior.</b></li> </ul>
Custom Policy (supported only in clusters of v1.15 or later)	<p><b>NOTA</b>                      Antes de establecer una política personalizada, debe instalar un complemento que admita la recopilación de métricas personalizadas en el clúster, por ejemplo, el complemento prometheus.</p> <ul style="list-style-type: none"> <li>● <b>Metric Name:</b> nombre de la métrica personalizada. Puede seleccionar un nombre como se le solicite. Para obtener más información, véase <a href="#">Monitoreo de métricas personalizadas con prometheus</a>.</li> <li>● <b>Metric Source:</b> Seleccione un tipo de objeto de la lista desplegable. Puede seleccionar <b>Pod</b>.</li> <li>● <b>Desired Value:</b> el valor métrico promedio de todos los pods. <math>\text{Número de pods a escalar (redondeado)} = (\text{Valor métrico actual} / \text{Valor deseado}) \times \text{Número de pods actuales}</math></li> </ul> <p><b>NOTA</b>                      Al calcular el número de pods que se agregarán o reducirán, la política de HPA utiliza el número máximo de pods en los últimos 5 minutos.</p> <ul style="list-style-type: none"> <li>● <b>Tolerance Range:</b> El escalado no se activa cuando el valor de la métrica está dentro del rango de tolerancia. El valor deseado debe estar dentro del rango de tolerancia.</li> </ul>

**Paso 4** Haga clic en **Create**.

----Fin

### 13.2.3 Creación de una política de CustomedHPA para el ajuste automático de cargas de trabajo

Una política de CustomedHPA escala las Deployments en función de métricas (como el uso de CPU y el uso de memoria) o en un intervalo periódico (un punto de tiempo específico cada día, cada semana, cada mes o cada año). Este tipo de política es una capacidad de ajuste automático mejorada por CCE.

Funciones soportadas:

- El ajuste se puede realizar basándose en el porcentaje del número actual de pods.
- Se puede establecer la etapa del ajuste mínimo.
- Se pueden realizar diferentes operaciones de ajuste según los valores métricos reales.

## Requisitos previos

Para usar una política de CustomedHPA, debe instalar el complemento **cce-hpa-controller**. Si la versión cce-hpa-controller es anterior a 1.2.11, el complemento **prometheus** debe estar instalado. Si la versión de cce-hpa-controller es 1.2.11 o posterior, los complementos que pueden proporcionar API de métricas deben estar instalados. Seleccione uno de los siguientes complementos según la versión del clúster y los requisitos reales.

- **metrics-server**: proporciona las métricas básicas de uso de recursos, como CPU de contenedor y uso de memoria. Es compatible con todas las versiones de clúster.
- **prometheus**: proporciona las métricas personalizadas además de las métricas básicas de recursos. Es necesario registrar Prometheus como el servicio que proporciona API de métricas. Para obtener más información, véase **Proporcionar métricas de recursos**. Este complemento solo admite clústeres de v1.21 o anteriores.
- **kube-prometheus-stack**: proporciona las métricas personalizadas además de las métricas básicas de recursos. Es necesario registrar Prometheus como el servicio que proporciona API de métricas. Para obtener más información, véase **Proporcionar métricas de recursos**. Este complemento solo admite clústeres de v1.23 o posterior.

## Notas y restricciones

- Las políticas de CustomedHPA solo se pueden crear para clústeres de v1.15 o posterior.
- Para los clústeres anteriores a v1.19.10, si se utiliza una política de HPA para escalar una carga de trabajo con volúmenes de EVS montados, los pods existentes no se pueden leer ni escribir cuando se programa un nuevo pod en otro nodo.  
  
Para los clústeres de v1.19.10 y de las versiones posteriores, si se utiliza una política de HPA para escalar una carga de trabajo con un volumen de EVS montado, no se puede iniciar un nuevo pod porque no se pueden conectar los discos de EVS.
- Las especificaciones de cce-hpa-controller se deciden por el número total de contenedores en el clúster y el número de políticas de ajuste. Se recomienda configurar 500m CPU y 1,000 MiB de memoria por cada 5,000 contenedores y 100m CPU y 500 MiB de memoria por cada 1,000 políticas de ajuste.

## Procedimiento

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** Elija **Workload Scaling** en el panel de navegación y haga clic en la ficha **CustomedHPA Policy**.

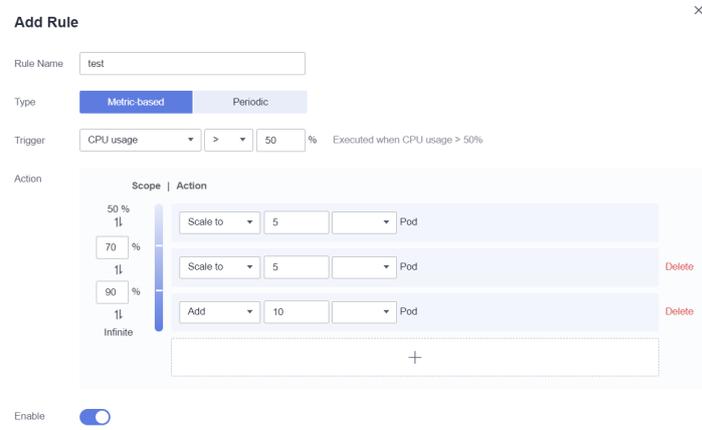
- Si aparece **Uninstalled** junto al nombre del complemento, haga clic en **Install** y configure los parámetros del complemento según sea necesario y haga clic en **Install** para instalar el complemento.
- Si aparece **Installed** junto al nombre del complemento, este complemento se ha instalado.

**Paso 3** Después de instalar el complemento, haga clic en **Create CustomedHPA Policy** en la esquina superior derecha.

**Paso 4** Establezca los parámetros de política.

**Tabla 13-5** Parámetros de política de CustomedHPA

Parámetro	Descripción
Policy Name	Nombre de la política que se va a crear. Configure este parámetro según sea necesario.
Namespace	Espacio de nombres al que pertenece la carga de trabajo.
Associated Workload	Carga de trabajo con la que está asociada la política de CustomedHPA.
Pod Range	Número mínimo y máximo de pods. Cuando se activa una política, los pods de carga de trabajo se escalan dentro de este intervalo.
Cooldown Period	Introduzca un intervalo, en minutos. Este parámetro indica el intervalo entre las operaciones de ajuste consecutivas. El período de tiempo de reutilización garantiza que se inicie una operación de ajuste solo cuando se haya completado la anterior y el sistema se esté ejecutando de manera estable.

Parámetro	Descripción
Rules	<p>Haga clic en . En el cuadro de diálogo que se muestra, establezca los siguientes parámetros:</p> <ul style="list-style-type: none"> <li>● <b>Name:</b> Introduzca un nombre de regla personalizado.</li> <li>● <b>Type:</b> Puede seleccionar <b>Metric-based</b> o <b>Periodic</b>.</li> </ul> <p><b>Metric-based:</b></p> <ul style="list-style-type: none"> <li>● <b>Trigger:</b> Seleccione <b>CPU usage</b> o <b>Memory usage</b>, elija &gt; o &lt; e introduzca un porcentaje. Como se muestra en la siguiente figura, la regla se ejecutará inmediatamente cuando el uso de CPU sea superior al 50%.</li> </ul> <p><b>NOTA</b>                      Uso = CPU o memorias utilizadas por pods/CPU o memorias solicitadas.</p> <ul style="list-style-type: none"> <li>● <b>Action:</b> Establezca una acción que se realizará cuando se cumpla la condición de activador. Se pueden agregar varias acciones. Como se muestra a continuación, cuando el uso de CPU supera el 50%, el número de pods se escala a 5. Cuando el uso de CPU supera el 70%, el número de pods se escala a 8. Cuando el uso de CPU supera el 90%, el número de pods se reduce a 18 (agregado 10 pods más). Estas reglas también funcionan para las operaciones de reducción.</li> <li>● <b>Enable:</b> Habilitar o deshabilitar la regla de política.</li> </ul> <p><b>Figura 13-2</b> Establecer una condición de activador</p>  <p><b>Periodic:</b></p> <ul style="list-style-type: none"> <li>● <b>Trigger Time:</b> Puede seleccionar un punto de tiempo específico cada día, cada semana, cada mes o cada año.</li> <li>● <b>Action:</b> Establezca una acción que se realizará cuando se alcance el valor <b>Triggered Time</b>. Como se muestra a continuación, se agregará un pod a las 17:00 todos los días.</li> <li>● <b>Enable:</b> Habilitar o deshabilitar la regla de política.</li> </ul>

Parámetro	Descripción
	<p><b>Figura 13-3</b> Activación periódica (Diario)</p> <p><b>Add Rule</b></p> <p>Rule Name <input type="text" value="test"/></p> <p>Type <span>Metric-based</span> <span><b>Periodic</b></span></p> <p>Trigger Time <input type="text" value="Every d..."/> <input type="text" value="17:00"/> </p> <p><small>This time indicates the local time of where the node is deployed.</small></p> <p>Action <input type="text" value="Add"/> <input type="text" value="1"/> <input type="text" value="Pod"/></p> <p>Enable <input checked="" type="checkbox"/></p> <p>Haga clic en <b>OK</b> y, a continuación, puede ver la regla agregada en la lista de políticas.</p>

**Paso 5** Haga clic en **Create**.

----Fin

## Uso de kubectl

Una política CustomHPA es un CustomResourceDefinition (CRD) y se puede definir de la siguiente manera en YAML:

```

apiVersion: autoscaling.cce.io/v1alpha1
kind: CustomizedHorizontalPodAutoscaler
metadata:
  name: customhpa-example
  namespace: default
spec:
  cooldownTime: 3m #Cooldown period
  maxReplicas: 10 # Maximum number of pods
  minReplicas: 1 # Minimum number of pods
  rules:
    - actions: #Policy rules
      - metricRange: 0,0.1 # Metric range, from 0 to 10%
        operationType: ScaleDown # Scaling type. ScaleDown indicates
        downsizing.
        operationUnit: Task #Operation unit. Task indicates the
        number of tasks.
        operationValue: 1 # Resource quantity in each scaling
      - metricRange: 0.1,0.3 # Metric range, from 10% to 30%
        operationType: ScaleDown
        operationUnit: Task
        operationValue: 2
    disable: false
  metricTrigger:
    hitThreshold: 1
    metricName: CPURatioToRequest # Metric name. CPURatioToRequest
    indicates the CPU usage.
    
```

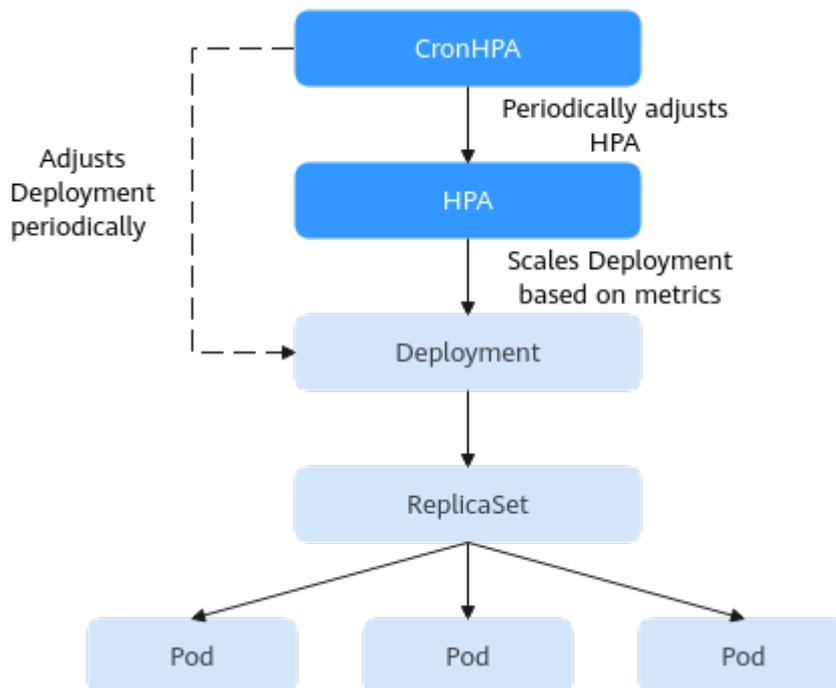
```

        metricOperation: <           # Metric expression operator
        metricValue: 0.3             # Value on the right of the metric
expression
    periodSeconds: 60               #
    statistic: instantaneous         #
    ruleName: low
    ruleType: Metric
    - actions:
      - metricRange: 0.7,0.9
        operationType: ScaleUp
        operationUnit: Task
        operationValue: 1
      - metricRange: 0.9,+Infinity
        operationType: ScaleUp
        operationUnit: Task
        operationValue: 2
    disable: false
    metricTrigger:
      hitThreshold: 1
      metricName: CPURatioToRequest
      metricOperation: '>'
      metricValue: 0.7
      periodSeconds: 60
      statistic: instantaneous
    ruleName: high
    ruleType: Metric
scaleTargetRef:                    # Associated workload
  apiVersion: apps/v1
  kind: Deployment
  name: nginx
    
```

## 13.2.4 Políticas de CronHPA

### Descripción general

Hay picos de tráfico predecibles e impredecibles para algunos servicios. Para estos servicios, CCE CronHPA le permite escalar recursos en los períodos fijos. Puede trabajar con las políticas de HPA para ajustar periódicamente el alcance de ajuste de HPA, implementando ajuste de carga de trabajo.



CronHPA puede ajustar periódicamente el número máximo y mínimo de pods en la política de HPA o ajustar directamente el número de pods de una Deployment.

El siguiente es un ejemplo YAML de CronHPA:

```
apiVersion: autoscaling.cce.io/v2alpha1
kind: CronHorizontalPodAutoscaler
metadata:
  name: ccetest
  namespace: default
spec:
  scaleTargetRef: # Associate an HPA policy or Deployment.
    apiVersion: autoscaling/v1
    kind: HorizontalPodAutoscaler
    name: hpa-test
  rules:
    - ruleName: "scale-down"
      schedule: "15 * * * *" # takes a Cron format string, for example, 0
      * * * * or @hourly.
      targetReplicas: 1 # Number of target pods
      disable: false
    - ruleName: "scale-up"
      schedule: "13 * * * *"
      targetReplicas: 6
      disable: false
```

**Tabla 13-6** Campos clave de CronHPA

Campo	Descripción
apiVersion	Versión de la API. El valor se fija en <b>autoscaling.cce.io/v2alpha1</b> .
kind	Tipo de la API. El valor se fija en <b>CronHorizontalPodAutoscaler</b> .
metadata.name	Nombre de una política de CronHPA.
metadata.namespace	Espacio de nombres al que pertenece la política CronHPA.
spec.scaleTargetRef	<p>Especifica el objeto de ajuste de CronHPA. Se pueden configurar los siguientes campos:</p> <ul style="list-style-type: none"> <li>● <b>apiVersion</b>: Versión de API del objeto de ajuste de CronHPA.</li> <li>● <b>kind</b>: Tipo de API del objeto de ajuste de CronHPA.</li> <li>● <b>name</b>: Nombre del objeto de ajuste de CronHPA.</li> </ul> <p>CronHPA admite políticas de HPA o Deployments. Para más detalles, consulte <a href="#">Uso de CronHPA para ajustar el alcance de escalado de HPA</a> o <a href="#">Uso de CronHPA para ajustar directamente el número de pods de Deployment</a>.</p>

Campo	Descripción
spec.rules	Regla de política de CronHPA. Se pueden agregar varias reglas. Se pueden configurar los siguientes campos para cada regla: <ul style="list-style-type: none"> <li>● <b>ruleName</b>: Nombre de regla de CronHPA, que debe ser único.</li> <li>● <b>schedule</b>: Tiempo de ejecución y período de un trabajo. Para obtener más información, consulte <a href="#">Cron</a>, por ejemplo, 0 * * * * o @hourly.</li> <li>● <b>targetReplicas</b>: indica el número de pods que se van a escalar dentro o fuera.</li> <li>● <b>disable</b>: El valor puede ser <b>true</b> o <b>false</b>. <b>false</b> indica que la regla entra en vigor, y <b>true</b> indica que la regla no entra en vigor.</li> </ul>

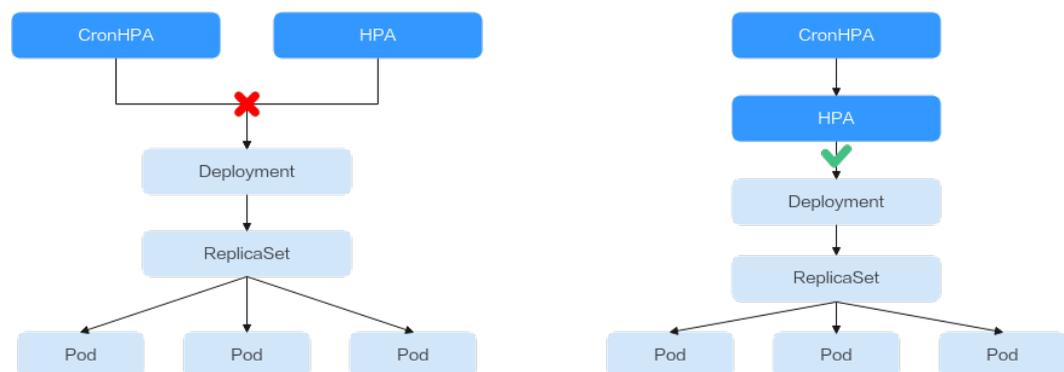
## Requisitos previos

Se ha instalado el complemento [cce-hpa-controller](#) de v1.2.13 o posterior.

## Uso de CronHPA para ajustar el alcance de escalado de HPA

CronHPA puede periódicamente ampliar/reducir los pods en las políticas de HPA para satisfacer servicios complejos.

HPA y CronHPA asocian objetos de ajuste mediante el campo **scaleTargetRef**. Si una Deployment es el objeto de ajuste tanto para CronHPA como para HPA, las dos políticas de ajuste son independientes entre sí. La operación realizada posteriormente sobrescribe la operación realizada anteriormente. Como resultado, el efecto de ajuste no cumple con las expectativas.



Cuando CronHPA es compatible con la política de HPA, el campo **scaleTargetRef** de CronHPA debe establecerse en la política de HPA y el campo **scaleTargetRef** de la política de HPA debe establecerse en Deployment. De esta manera, CronHPA ajusta el número máximo y mínimo de pods en la política de HPA en un tiempo fijo y el ajuste programado es compatible con el ajuste automático.

### Paso 1 Cree una política de HPA para la Deployment.

```
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: hpa-test
```

```

namespace: default
spec:
  maxReplicas: 10           # Maximum number of pods
  minReplicas: 5           # Minimum number of pods
  scaleTargetRef:         # Associate a Deployment.
    apiVersion: apps/v1
    kind: Deployment
    name: nginx
  targetCPUUtilizationPercentage: 50
    
```

**Paso 2** Cree una política de CronHPA y asíciela con la política de HPA creada en [Paso 1](#).

```

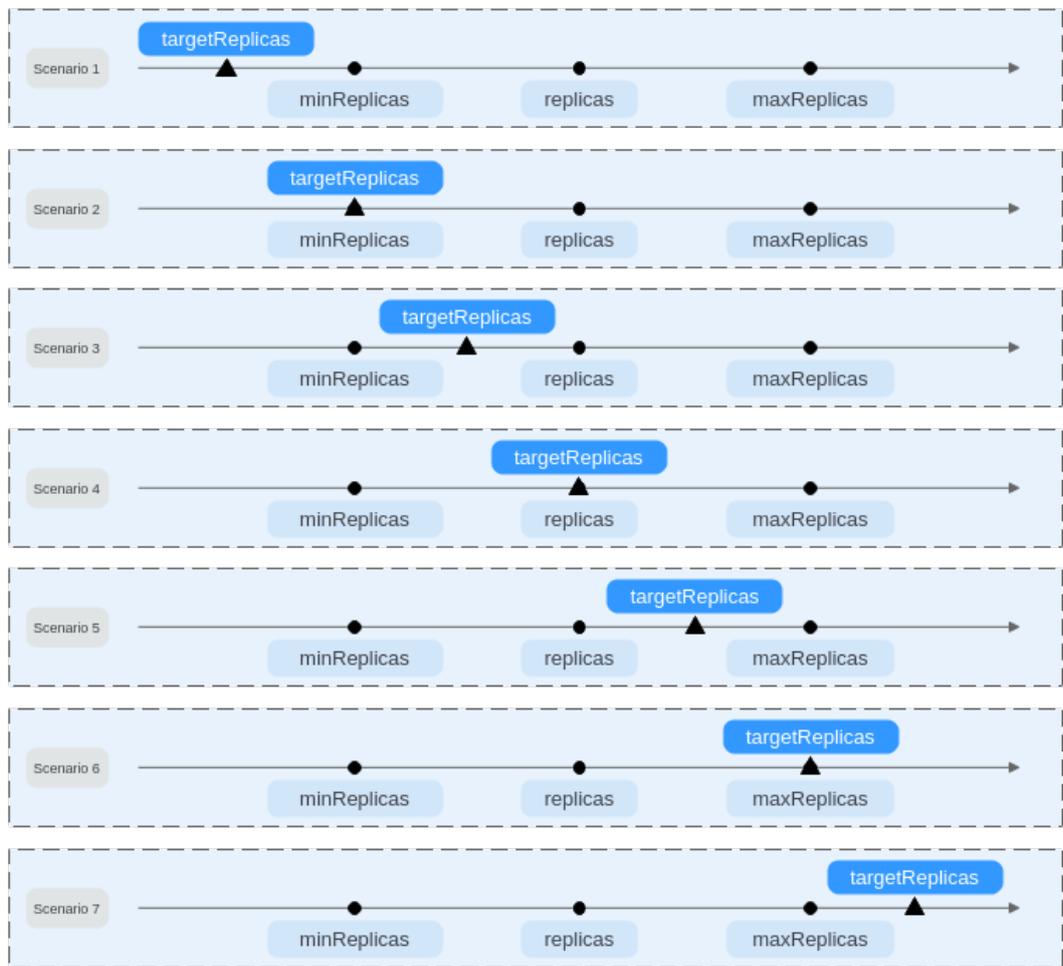
apiVersion: autoscaling.cce.io/v2alpha1
kind: CronHorizontalPodAutoscaler
metadata:
  name: ccetest
  namespace: default
spec:
  scaleTargetRef:         # Associate the HPA policy
    apiVersion: autoscaling/v1
    kind: HorizontalPodAutoscaler
    name: hpa-test
  rules:
  - ruleName: "scale-down"
    schedule: "15 * * * *" # Running time and period of a job. For
    # details, see Cron, for example, 0 * * * * and @hourly.
    targetReplicas: 1      # Number of target pods
    disable: false
  - ruleName: "scale-up"
    schedule: "13 * * * *"
    targetReplicas: 11
    disable: false
    
```

Cuando CronHPA y HPA se usan juntos, las reglas de CronHPA entran en vigor según la política de HPA. CronHPA utiliza HPA para realizar operaciones en la Deployment. Comprender los siguientes parámetros puede comprender mejor el principio de funcionamiento del CronHPA.

- **targetReplicas**: Número de pods fijados para CronHPA. Cuando CronHPA entra en vigor, este parámetro ajusta el número máximo o mínimo de pods en las políticas de HPA para ajustar el número de pods de Deployment.
- **minReplicas**: Número mínimo de pods de Deployment.
- **maxReplicas**: Número máximo de pods de Deployment.
- **replicas**: Número de pods en una Deployment antes de que la política de CronHPA entre en vigor.

Cuando la regla de CronHPA entra en vigor, el número máximo o mínimo de pods se ajusta comparando el número de **targetReplicas** con el número real de pods y combinando el número mínimo o máximo de pods en la política de HPA.

**Figura 13-4** Escenarios de ajuste de CronHPA



**Figura 13-4** muestra posibles escenarios de ajuste. Los siguientes ejemplos detallan cómo CronHPA modifica el número de pods en HPA.

Escenario	Descripción del escenario	Cronhpa (targetReplicas)	Deployment (replicas)	HPA (minReplicas / maxReplicas)	Resultados	Descripción de operaciones
Escenario 1	$\text{targetReplicas} < \text{minReplicas} \leq \text{replicas} \leq \text{maxReplicas}$	4	5	5/10	HPA: 4/10 Deployment: 5	Cuando el valor de <b>targetReplicas</b> es menor que el de <b>minReplicas</b> : <ul style="list-style-type: none"> <li>● Cambie el valor de <b>minReplicas</b>.</li> <li>● El valor de <b>replicas</b> no requiere ningún cambio.</li> </ul>

Escenario	Descripción del escenario	Cronhpa (target Replicas)	Deployment (replicas)	HPA (minReplicas / maxReplicas)	Resultados	Descripción de operaciones
Escenario 2	$\text{targetReplicas} = \text{minReplicas} \leq \text{replicas} \leq \text{maxReplicas}$	5	6	5/10	HPA: 5/10 Deployment: 6	Cuando el valor de <b>targetReplicas</b> es menor que el de <b>minReplicas</b> : <ul style="list-style-type: none"> <li>● El valor de <b>minReplicas</b> no requiere ningún cambio.</li> <li>● El valor de <b>replicas</b> no requiere ningún cambio.</li> </ul>
Escenario 3	$\text{minReplicas} < \text{targetReplicas} < \text{replicas} \leq \text{maxReplicas}$	4	5	1/10	HPA: 4/10 Deployment: 5	Cuando el valor de <b>targetReplicas</b> es mayor que el de <b>minReplicas</b> y menor que el de <b>replicas</b> : <ul style="list-style-type: none"> <li>● Cambie el valor de <b>minReplicas</b>.</li> <li>● El valor de <b>replicas</b> no requiere ningún cambio.</li> </ul>
Escenario 4	$\text{minReplicas} < \text{targetReplicas} = \text{replicas} < \text{maxReplicas}$	5	5	1/10	HPA: 5/10 Deployment: 5	Cuando el valor de <b>targetReplicas</b> es mayor que el de <b>minReplicas</b> e igual al de <b>replicas</b> : <ul style="list-style-type: none"> <li>● Cambie el valor de <b>minReplicas</b>.</li> <li>● El valor de <b>replicas</b> no requiere ningún cambio.</li> </ul>

Escenario	Descripción del escenario	Cronpa (targetReplicas)	Deployment (replicas)	HPA (minReplicas / maxReplicas)	Resultados	Descripción de operaciones
Escenario 5	$\text{minReplicas} \leq \text{replicas} < \text{targetReplicas} < \text{maxReplicas}$	6	5	1/10	HPA: 6/10 Deployment: 6	Cuando el valor de <b>targetReplicas</b> es mayor que el de <b>replicas</b> y menor que el de <b>maxReplicas</b> : <ul style="list-style-type: none"> <li>● Cambie el valor de <b>minReplicas</b>.</li> <li>● Cambie el valor de <b>replicas</b>.</li> </ul>
Escenario 6	$\text{minReplicas} \leq \text{replicas} < \text{targetReplicas} = \text{maxReplicas}$	10	5	1/10	HPA: 10/10 Deployment: 10	Cuando el valor de <b>targetReplicas</b> es mayor que el de <b>replicas</b> e igual al de <b>maxReplicas</b> : <ul style="list-style-type: none"> <li>● Cambie el valor de <b>minReplicas</b>.</li> <li>● Cambie el valor de <b>replicas</b>.</li> </ul>
Escenario 7	$\text{minReplicas} \leq \text{replicas} \leq \text{maxReplicas} < \text{targetReplicas}$	11	5	5/10	HPA: 11/11 Deployment: 11	Cuando el valor de <b>targetReplicas</b> es mayor que el de <b>maxReplicas</b> : <ul style="list-style-type: none"> <li>● Cambie el valor de <b>minReplicas</b>.</li> <li>● Cambie el valor de <b>maxReplicas</b>.</li> <li>● Cambie el valor de <b>replicas</b>.</li> </ul>

----Fin

## Uso de CronHPA para ajustar directamente el número de pods de Deployment

CronHPA ajusta las implementaciones asociadas por separado para ajustar periódicamente el número de pods de Deployment. El método es el siguiente:

```
apiVersion: autoscaling.cce.io/v2alpha1
kind: CronHorizontalPodAutoscaler
metadata:
  name: cctest
  namespace: default
spec:
```

```

scaleTargetRef:          # Associate a Deployment.
  apiVersion: apps/v1
  kind: Deployment
  name: nginx
rules:
- ruleName: "scale-down"
  schedule: "08 * * * *" # Running time and period of a job. For details,
see Cron, for example, 0 * * * * or @hourly.
  targetReplicas: 1
  disable: false
- ruleName: "scale-up"
  schedule: "05 * * * *"
  targetReplicas: 3
  disable: false
    
```

## 13.2.5 Gestión de políticas de escalado de carga de trabajo

### Escenario

Después de crear una política de HPA o de CustomedHPA, puede actualizar, clonar, editar y eliminar la política, así como editar el archivo YAML.

### Comprobación de una política de HPA

Puede ver las reglas, el estado y los eventos de una política de HPA y controlar las excepciones según la información de error mostrada.

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** En el panel de navegación, elija **Workload Scaling**. En la página de la ficha **HPA Policies**, haga clic en  junto a la política de HPA de destino.

**Paso 3** En el área expandida, puede ver las páginas de fichas **Rules**, **Status** y **Events**. Si la política es anormal, localice y rectifique el error basándose en la información de error.

#### **NOTA**

También puede ver la política de HPA creada en la página de detalles de la carga de trabajo.

1. Inicie sesión en la consola de CCE y acceda a la consola del clúster.
2. En el panel de navegación, elija **Workloads**. Haga clic en el nombre de la carga de trabajo para ver sus detalles.
3. En la página de detalles de la carga de trabajo, cambie a la página de ficha **Auto Scaling** para ver las políticas de HPA o de CustomedHPA. También puede ver las políticas de ajuste configuradas en **Workload Scaling**.

**Tabla 13-7** Tipos y nombres de eventos

Tipo de evento	Nombre del evento	Descripción
Normal	SuccessfulRescale	El ajuste se realiza correctamente.
Anormal	InvalidTargetRange	Rango de destino no válido.
	InvalidSelector	Selector no válido.
	FailedGetObjectMetric	Los objetos no se obtienen.

Tipo de evento	Nombre del evento	Descripción
	FailedGetPodsMetric	Los pods no se obtienen.
	FailedGetResourceMetric	Los recursos no se obtienen.
	FailedGetExternalMetric	Las métricas externas no se obtienen.
	InvalidMetricSourceType	Tipo de origen de métrica no válido.
	FailedConvertHPA	Error en la conversión de HPA.
	FailedGetScale	La escala no se obtiene.
	FailedComputeMetricsReplicas	Error al calcular réplicas definidas por métricas.
	FailedGetScaleWindow	Error al obtener ScaleWindow.
	FailedRescale	Error al escalar el servicio.

----Fin

## Consulta de una política de CustomedHPA

Puede ver las reglas y el estado más reciente de una política de CustomedHPA y rectificar errores basándose en la información de error mostrada.

- Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.
- Paso 2** En el panel de navegación, elija **Workload Scaling**. En la página de la ficha **CustomHPA Policies**, haga clic en  junto a la política de CustomHPA de destino.
- Paso 3** En el área expandida, si la política es anormal en la página de ficha **Rules**, haga clic en **Details** de **Latest Status** y busque el error en función de la información mostrada.

### NOTA

También puede ver la política de HPA creada en la página de detalles de la carga de trabajo.

1. Inicie sesión en la consola de CCE y acceda a la consola del clúster.
2. En el panel de navegación, elija **Workloads**. Haga clic en el nombre de la carga de trabajo para ver sus detalles.
3. En la página de detalles de la carga de trabajo, cambie a la página de ficha **Auto Scaling** para ver las políticas de HPA o de CustomedHPA. También puede ver las políticas de ajuste configuradas en **Workload Scaling**.

----Fin

## Actualización de una política de HPA o de CustomedHPA

Una política de HPA se utiliza como ejemplo.

- Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

- Paso 2** En el panel de navegación, elija **Workload Scaling**. Haga clic en **Update** en la columna **Operation** de la política de destino.
- Paso 3** En la página **Update HPA Policy** que se muestra, establezca los parámetros de política que aparecen en **Tabla 13-4**.
- Paso 4** Haga clic en **Update**.

----Fin

## Edición del archivo YAML (política de HPA)

- Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.
- Paso 2** En el panel de navegación, elija **Workload Scaling**. Haga clic en **More > Edit YAML** en la columna **Operation** de la política de HPA de destino.
- Paso 3** En el cuadro de diálogo **Edit YAML** que se muestra, edite o descargue el archivo YAML.
- Paso 4** Haga clic en el botón de cerrar en la esquina superior derecha.

----Fin

## Consulta del archivo YAML (Política de CustomedHPA)

- Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.
- Paso 2** En el panel de navegación, elija **Workload Scaling**. Haga clic en **View YAML** en la columna **Operation** de la política de CustomedHPA de destino.
- Paso 3** En el cuadro de diálogo que se muestra, puede copiar y descargar el archivo YAML, pero no puede modificarlo.
- Paso 4** Haga clic en el botón de cerrar en la esquina superior derecha.

----Fin

## Eliminación de una política de HPA o de CustomedHPA

- Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.
- Paso 2** En el panel de navegación, elija **Workload Scaling**. Haga clic en **Delete** en la columna **Operation** de la política de destino.
- Paso 3** En el cuadro de diálogo que aparece, haga clic en **Yes**.

----Fin

# 13.3 Ajuste de un nodo

## 13.3.1 Mecanismos de escalado de nodos

Kubernetes HPA está diseñado para pods. Sin embargo, si los recursos del clúster son insuficientes, solo puede agregar nodos. El escalado de los nodos de clúster podría ser laborioso. Ahora, con las nubes, puede agregar o eliminar nodos simplemente invocando a la API.

**autoscaler** es un componente proporcionado por Kubernetes para el ajuste automático de nodos de clúster en función del estado de programación de pods y el uso de recursos.

## Requisitos previos

Antes de usar la función de ajuste de nodos, debe instalar el complemento de **autoscaler** de v1.13.8 o posterior.

## Cómo funciona autoscaler

autoscaler pasa por dos procesos.

- Expansión horizontal: autoscaler comprueba todos los pods no programados cada 10 segundos y selecciona un grupo de nodos que cumpla con los requisitos de expansión según la política que establezcas.
- Reducción horizontal: autoscaler escanea todos los nodos cada 10 segundos. Si el número de solicitudes de pod en un nodo es menor que el porcentaje definido por el usuario para reducción, autoscaler simula si los pods en el nodo se pueden migrar a otros nodos. En caso afirmativo, el nodo se eliminará después de una ventana de tiempo inactivo.

Cuando un nodo de clúster está inactivo durante un período de tiempo (10 minutos de forma predeterminada), se activa la ampliación del clúster y el nodo se elimina automáticamente. Sin embargo, no se puede eliminar un nodo de un clúster si existen los siguientes pods:

- Pods que no cumplen con los requisitos específicos establecidos en Pod Disruption Budgets (**PodDisruptionBudget**)
- Pods que no se pueden programar para otros nodos debido a restricciones como las políticas de afinidad y antiafinidad
- Pods que tienen la anotación **cluster-autoscaler.kubernetes.io/safe-to-evict: 'false'**
- Pods (excepto aquellos creados por DaemonSets en el espacio de nombres del sistema kube) que existen en el espacio de nombres del sistema kube en el nodo
- Pods que no son creados por el controlador (Deployment/ReplicaSet/job/StatefulSet)

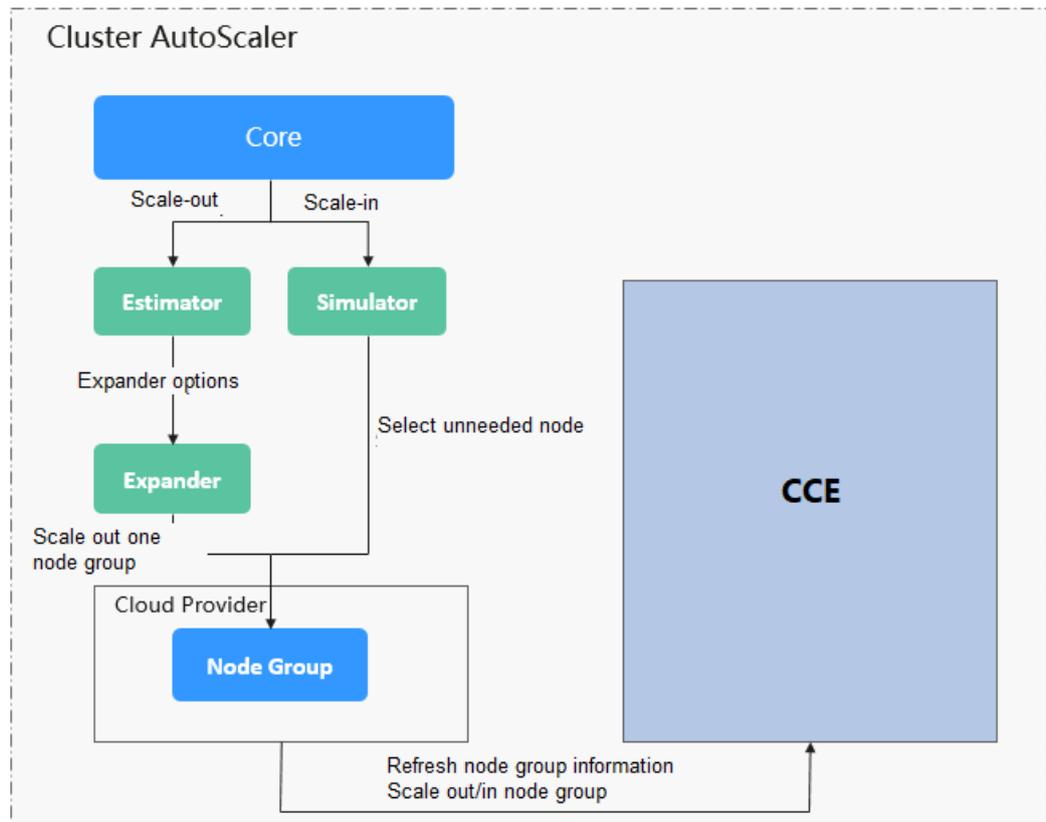
### NOTA

Cuando un nodo cumple con las condiciones de la reducción, el autoscaler agrega la mancha **DeletionCandidateOfClusterAutoscaler** al nodo por adelantado para evitar que los pods se programen en el nodo. Después de desinstalar el complemento de autoscaler, si la mancha todavía existe en el nodo, elimínelo manualmente.

## Arquitectura de autoscaler

**Figura 13-5** muestra la arquitectura del autoscaler y sus módulos principales:

Figura 13-5 Arquitectura de autoscaler



### Descripción

- **Estimator:** Evalúa el número de nodos que se agregarán a cada grupo de nodos para alojar pods no programados.
- **Simulator:** Busca los nodos que cumplen las condiciones reducción en el escenario reducción.
- **Expander:** Selecciona un nodo óptimo del grupo de nodos seleccionado por el Estimador en función de la política definida por el usuario en el escenario de la expansión. Actualmente, el Expander tiene las siguientes políticas:
  - **Random:** Se selecciona aleatoriamente un grupo de nodos.
  - **most-Pods:** Selecciona el grupo de nodos que puede alojar el mayor número de pods no programados después de la expansión. Si varios grupos de nodos cumplen con el requisito, se seleccionará un grupo de nodos aleatorios.
  - **least-waste:** Selecciona el grupo de nodos que tiene el menor desperdicio de recursos de CPU o memoria después de la función expansión.
  - **price:** Selecciona el grupo de nodos en el que los nodos a agregar cuestan menos expansión.
  - **priority:** Selecciona el grupo de nodos con la ponderación más alta. Las ponderaciones son definidas por el usuario.

Actualmente, CCE es compatible con todas las políticas excepto **price**. De forma predeterminada, los complementos de CCE utilizan la política **least-waste**.

## 13.3.2 Creación de una política del ajuste de nodos

CCE proporciona el ajuste automático a través del complemento del **autoscaler**. Los nodos con diferentes especificaciones se pueden agregar automáticamente a través de AZ bajo demanda.

Si una política de ajuste de nodos y la configuración en el complemento de autoscaler tienen efecto al mismo tiempo, por ejemplo, hay pods que no se pueden programar y el valor de una métrica alcanza el umbral al mismo tiempo, la expansión horizontal se realiza primero para los pods no programados.

- Si la expansión horizontal tiene éxito para los pods no programables, el sistema omite la lógica de reglas basada en métricas y entra en el siguiente bucle.
- Si la expansión horizontal falla para los pods no programables, se ejecuta la regla basada en métricas.

### Requisitos previos

Antes de utilizar la función de ajuste de nodos, debe instalar el complemento de **autoscaler** de v1.13.8 o posterior en el clúster.

### Notas y restricciones

- Solo los pools de nodo de pago por uso admiten el ajuste automático.
- Las políticas de ajuste automático se aplican a los grupos de nodos. Cuando el número de nodos en un grupo de nodos es 0 y la política de ajuste se basa en el uso de CPU o memoria, el ajuste de nodos no se activa.
- La reducción del nodo causará la pérdida de datos de PVC/PV para los **PV locales** asociados con el nodo. Estos PVC y PV no se pueden restaurar o utilizar de nuevo. En una reducción de nodo, el pod que utiliza el PV local se desaloja del nodo. Se crea un nuevo pod y permanece en el estado pendiente. Esto se debe a que el PVC utilizado por el pod tiene una etiqueta de nodo, debido a lo cual el pod no se puede programar.

### Procedimiento

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** Elija **Node Scaling** en el panel de navegación.

- Si aparece **Uninstalled** junto al nombre del complemento, haga clic en **Install** y configure los parámetros del complemento según sea necesario y haga clic en **Install** para instalar el complemento.
- Si aparece **Installed** junto al nombre del complemento, este complemento se ha instalado.

**Paso 3** Haga clic en **Create Node Scaling Policy** en la esquina superior derecha y establezca los parámetros de la siguiente manera:

- **Policy Name:** nombre de la política que se va a crear, que se puede personalizar.
- **Associated Node Pools:** Seleccione el grupo de nodos que se va a asociar. Puede asociar varios grupos de nodos para utilizar la misma política de ajuste.
- **Rules:** Haga clic en **Add Rule**. En el cuadro de diálogo que se muestra, establezca los siguientes parámetros:
  - Rule Name:** Ingrese un nombre de regla.

**Rule Type:** Puede seleccionar **Metric-based** o **Periodic**. Las diferencias entre los dos tipos son las siguientes:

– **Metric-based:**

**Condition:** Seleccione **CPU allocation rate** o **Memory allocation rate** e introduzca un valor. El valor debe ser mayor que el porcentaje de reducción configurado en el complemento del autoscaler.

 **NOTA**

- Asignación de recursos (%) = Recursos solicitados por pods en el grupo de nodos/ Recursos asignados a pods en el grupo de nodos
- **Si varias reglas cumplen las condiciones, las reglas se ejecutan en cualquiera de los siguientes modos:**
  - Si se configuran reglas basadas en **CPU allocation rate** y **memory allocation rate** y dos o más reglas cumplen las condiciones de la expansión horizontal, se ejecutará la regla que agregará la mayoría de los nodos.
  - Si se configura una regla basada en **CPU allocation rate** y a **periodic rule** y ambas cumplen con las condiciones de la expansión horizontal, una de ellas se ejecutará aleatoriamente. La regla ejecutada primero (regla A) cambia el grupo de nodos al estado de ajuste. Como resultado, la otra regla (regla B) no se puede ejecutar. Después de ejecutar la regla A y de que el estado del grupo de nodos se vuelva normal, la regla B no se ejecutará.
- Si se configuran reglas basadas en **CPU allocation rate** y **memory allocation rate**, el período de detección de políticas varía con la lógica de procesamiento de cada bucle del complemento de autoscaler. La expansión horizontal se activa una vez que se cumplen las condiciones, pero está limitada por otros factores tales como el intervalo de enfriamiento y el estado del grupo de nodos.

– **Periodic:**

**Trigger Time:** Puede seleccionar un punto de tiempo específico cada día, cada semana, cada mes o cada año.

**Action:** Establezca una acción que se realizará cuando se cumpla la condición de activador.

Puede hacer clic de nuevo en **Add Rule** para agregar más políticas de ajuste de nodos. Puede agregar un máximo de una regla basada en el uso de CPU y una regla basada en el uso de memoria. El número total de reglas no puede exceder de 10.

**Paso 4** Haga clic en **OK**.

----Fin

## Restricciones en la reducción horizontal

Solo puede establecer políticas de reducción de nodo cuando instale el [complemento de autoscaler](#).

La reducción horizontal de nodo solo puede activarse mediante la tasa de asignación de recursos. Cuando las velocidades de asignación de CPU y memoria en un clúster son inferiores a los umbrales especificados (establecidos cuando se instala o modifica el complemento del autoscaler), se activa la reducción horizontal para los nodos del grupo de nodos (esta función se puede deshabilitar), como se muestra en [Figura 13-6](#).

**Figura 13-6** Configuración automática de la reducción horizontal

**Parameters**

Scaling  Nodes are automatically added (from the node pool) when pods in the cluster cannot be scheduled.  
 Auto node scale-in

Node Idle Time (min)  Minute  
How long a node should be unneeded before it is eligible for scale down. The default value is 10 minutes.

Scale-in Threshold  %  
When the resource usage of a node is lower than a specified value (percentage), the node is considered idle (both CPUs and memory need to meet the requirements).

Stabilization Window (s)  Minute  
How long after a scale-out that a scale-in evaluation resumes.

Minute  
How long after the node deletion that a scale-in evaluation resumes.

Minute  
How long after a scale-in failure that a scale-in evaluation resumes.

Max. Nodes for Batch Deletion   
Maximum number of empty nodes that can be deleted at the same time.

Check Interval  Minute  
Interval for checking again a node that could not be removed before.

## Ejemplo de YAML

A continuación se muestra un ejemplo de YAML de una política de ajuste de nodos:

```
apiVersion: autoscaling.cce.io/v1alpha1
kind: HorizontalNodeAutoscaler
metadata:
  creationTimestamp: "2020-02-13T12:47:49Z"
  generation: 1
  name: xxxx
  namespace: kube-system
  resourceVersion: "11433270"
  selfLink: /apis/autoscaling.cce.io/v1alpha1/namespaces/kube-system/horizontalnodeautoscalers/xxxx
  uid: c2bd1e1d-60aa-47b5-938c-6bf3fadbe91f
spec:
  disable: false
  rules:
  - action:
    type: ScaleUp
    unit: Node
    value: 1
    cronTrigger:
      schedule: 47 20 * * *
    disable: false
    ruleName: cronrule
    type: Cron
  - action:
    type: ScaleUp
    unit: Node
```

```

value: 2
disable: false
metricTrigger:
  metricName: Cpu
  metricOperation: '>'
  metricValue: "40"
  unit: Percent
ruleName: metricrule
type: Metric
targetNodepoolIds:
- 7d48eca7-3419-11ea-bc29-0255ac1001a8
    
```

**Tabla 13-8** Parámetros de clave

Parámetro	Tipo	Descripción
spec.disable	Bool	Si se debe habilitar la política de ajuste. Este parámetro tiene efecto para todas las reglas de la política.
spec.rules	Array	Todas las reglas de una política de ajuste.
spec.rules[x].ruleName	String	Nombre de la regla.
spec.rules[x].type	String	Tipo de la regla. Actualmente, son compatibles con <b>Cron</b> y <b>Metric</b> .
spec.rules[x].disable	Bool	Cambio de la regla. Actualmente, solo se admite <b>false</b> .
spec.rules[x].action.type	String	Tipo de acción de regla. Actualmente, solo se admite <b>ScaleUp</b> .
spec.rules[x].action.unit	String	Unidad de acción de reglas. Actualmente, solo se admite <b>Node</b> .
spec.rules[x].action.value	Integer	Valor de acción de regla.
spec.rules[x].cronTrigger	/	Opcional. Este parámetro solo es válido en las reglas periódicas.
spec.rules[x].cronTrigger.schedule	String	Expresión de Cron de una regla periódica.
spec.rules[x].metricTrigger	/	Opcional. Este parámetro solo es válido en reglas basadas en métricas.
spec.rules[x].metricTrigger.metricName	String	Métrica de una regla basada en métricas. Actualmente, son compatibles con <b>Cpu</b> y <b>Memory</b> .
spec.rules[x].metricTrigger.metricOperation	String	Operador de comparación de una regla basada en métricas. Actualmente, solo se admite <b>&gt;</b> .

Parámetro	Tipo	Descripción
spec.rules[x].metricTrigger.metricValue	String	Umbral métrico de una regla basada en métricas. El valor puede ser cualquier entero de 1 a 100 y debe ser una cadena de caracteres.
spec.rules[x].metricTrigger.Unit	String	Unidad del umbral de regla basado en métricas. Actualmente, solo se admite %.
spec.targetNodepoolIds	Array	Todos los grupos de nodos asociados a la política de ajuste.
spec.targetNodepoolIds[x]	String	ID del grupo de nodos asociado a la política de ajuste.

### 13.3.3 Gestión de políticas de escalado de nodos

#### Escenario

Después de crear una política de ajuste de nodos, puede eliminar, editar, deshabilitar, habilitar o clonar la política.

#### Consulta de una política de escalado de nodos

Puede ver el grupo de nodos asociado, las reglas y el historial de ajuste de una política de ajuste de nodos y rectificar errores de acuerdo con la información de error mostrada.

- Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.
- Paso 2** Elija **Node Scaling** en el panel de navegación y haga clic en  delante de la política que se va a ver.
- Paso 3** En el área expandida, se muestran las páginas de fichas **Associated Node Pools**, **Rules** y **Scaling History**. Si la política es anormal, localice y rectifique el error basándose en la información de error.

#### NOTA

También puede deshabilitar o habilitar el ajuste automático en la página **Node Pools**.

1. Inicie sesión en la consola de CCE y acceda a la consola del clúster.
2. En el panel de navegación, elija **Nodes** y cambie a la página de ficha **Node Pools**.
3. Haga clic en **Update** del grupo de nodos que se va a operar. En el cuadro de diálogo **Update Node Pool** que se muestra, establezca los límites del número de nodos y el período de enfriamiento.

----Fin

#### Eliminación de una política de escalado de nodos

- Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** Elija **Node Scaling** en el panel de navegación y elija **More > Delete** junto a la política que se va a eliminar.

**Paso 3** En el cuadro de diálogo **Delete Node Scaling Policy** que se muestra, confirme si desea eliminar la política.

**Paso 4** Haga clic en **Yes** para eliminar la política.

----Fin

## Edición de una política de escalado de nodos

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** Elija **Node Scaling** en el panel de navegación y haga clic en **Edit** en la columna **Operation** de la política que se va a editar.

**Paso 3** En la página **Edit Node Scaling Policy** mostrada, modifique los valores de los parámetros de política que aparecen en la lista de [Tabla 13-8](#).

**Paso 4** Una vez completada la configuración, haga clic en **OK**.

----Fin

## Clonar una política de escalado de nodos

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** Elija **Node Scaling** en el panel de navegación y elija **More > Clone** junto a la política que se va a clonar.

**Paso 3** En la página **Clone Node Scaling Policy** mostrada, se han clonado ciertos parámetros. Agregar o modificar otros parámetros de política en función de los requisitos de servicio.

**Paso 4** Haga clic en **OK**.

----Fin

## Habilitar o deshabilitar una política de escalado de nodos

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster.

**Paso 2** Elija **Node Scaling** en el panel de navegación y haga clic en **Disable** en la columna **Operation** de la política que se va a deshabilitar. Si la política está deshabilitada, haga clic en **Enable** en la columna **Operation** de la política.

**Paso 3** En el cuadro de diálogo que se muestra, confirme si desea deshabilitar o habilitar la política de nodo.

----Fin

## 13.4 Uso de HPA y CA para el ajuste automático de cargas de trabajo y nodos

### Escenarios de aplicación

La mejor manera de manejar el tráfico creciente es ajustar automáticamente el número de máquinas según el volumen de tráfico o el uso de recursos, lo que se denomina ajuste.

Cuando se utilizan pods o contenedores para desplegar aplicaciones, normalmente se requiere establecer el límite superior de recursos disponibles para pods o contenedores para evitar el uso ilimitado de recursos de nodo durante las horas pico. Sin embargo, después de alcanzar el límite superior, puede producirse un error de aplicación. Para resolver este problema, escala el edición de pods para compartir cargas de trabajo. Si el uso de recursos de nodo aumenta hasta cierto punto que los pods recién agregados no pueden planificarse, ajuste el número de nodos basándose en el uso de recursos de nodo.

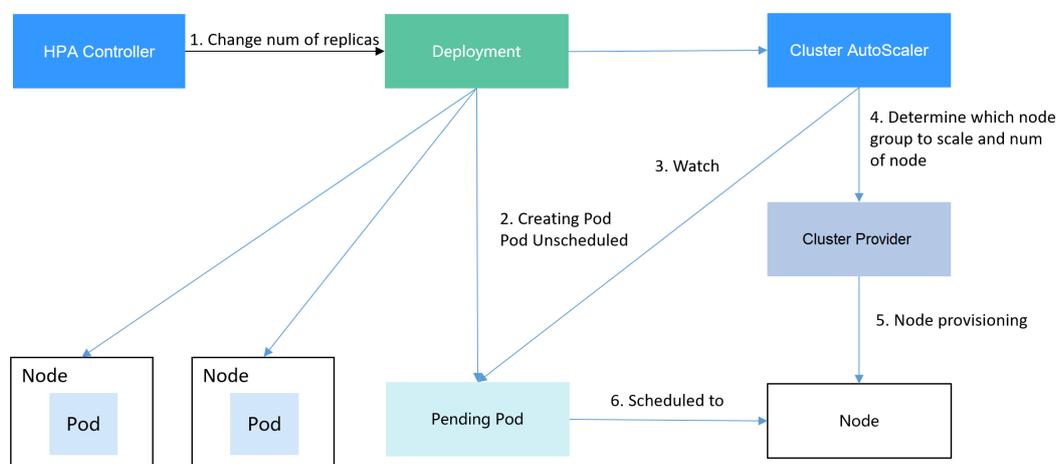
### Solución

Dos políticas principales de ajuste automático son HPA (Horizontal Pod Autoscaling) y CA (Cluster AutoScaling). HPA es para el ajuste automático de la carga de trabajo y CA es para el ajuste automático de nodos.

HPA y CA trabajan entre sí. HPA requiere suficientes recursos de clúster para escalar con éxito. Cuando los recursos del clúster son insuficientes, se necesita CA para agregar nodos. Si HPA reduce las cargas de trabajo, el clúster tendrá una gran cantidad de recursos inactivos. En este caso, CA necesita liberar nodos para evitar el desperdicio de recursos.

Como se muestra en [Figura 13-7](#), HPA realiza expansión según las métricas de monitorización. Cuando los recursos del clúster son insuficientes, los pods recién creados están en estado Pending. A continuación, CA comprueba estos pods pendientes y selecciona el grupo de nodos más apropiado según la política de ajuste configurada para escalar el grupo de nodos. Para obtener detalles sobre cómo funcionan HPA y CA, consulte [Mecanismos de ajuste de la carga de trabajo](#) y [Mecanismos de ajuste de los nodos](#).

**Figura 13-7** Flujos de trabajo de HPA y CA

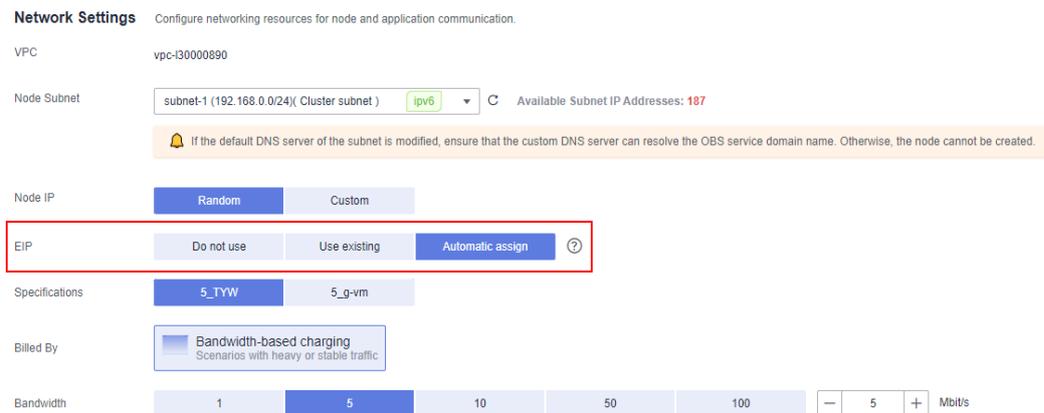


El uso de HPA y CA puede implementar fácilmente el ajuste automático en la mayoría de los escenarios. Además, el proceso de ajuste de nodos y pods se puede observar fácilmente.

En esta sección se utiliza un ejemplo para describir el proceso de ajuste automático con políticas de HPA y CA juntas.

## Preparaciones

**Paso 1** Cree un clúster con un nodo. El nodo debe tener 2 núcleos de vCPUs y 4 GiB de memoria, o una especificación más alta, así como una EIP para permitir el acceso externo. Si no hay ninguna EIP vinculada al nodo durante la creación del nodo, puede enlazar manualmente uno en la consola de ECS después de crear el nodo.



**Paso 2** Instalar complementos para el clúster.

- Mecanismos de escalado: complemento de ajuste de nodos
- metrics-server: un agregador de datos de uso de recursos en un clúster de Kubernetes. Puede recopilar datos de medición de los principales recursos de Kubernetes, como pods, nodos, contenedores y Services.

**Paso 3** Inicie sesión en el nodo del clúster y ejecute una aplicación informática intensiva. Cuando un usuario envía una solicitud, el resultado debe calcularse antes de ser devuelto al usuario.

1. Cree un archivo de PHP llamado **index.php** para calcular la raíz cuadrada de la solicitud de 1,000,000 veces antes de devolver **OK!**.

```
vi index.php
```

El contenido del archivo es el siguiente:

```
<?php
    $x = 0.0001;
    for ($i = 0; $i <= 1000000; $i++) {
        $x += sqrt($x);
    }
    echo "OK!";
?>
```

2. Compile un archivo **Dockerfile** para crear una imagen.

```
vi Dockerfile
```

El contenido es el siguiente:

```
FROM php:5-apache
COPY index.php /var/www/html/index.php
RUN chmod a+rx index.php
```

3. Ejecute el siguiente comando para crear una imagen llamada **hpa-example** con la etiqueta **latest**.

```
docker build -t hpa-example:latest .
```

- (Opcional) Inicie sesión en la consola de SWR, elija **Organizations** en el panel de navegación y haga clic en **Create Organization** en la esquina superior derecha para crear una organización.

Omita este paso si ya tiene una organización.

- En el panel de navegación, elija **My Images** y, a continuación, haga clic en **Upload Through Client**. En la página mostrada, haga clic en **Generate a temporary login command** y haga clic en  para copiar el comando.
- Ejecute el comando login copiado en el paso anterior en el nodo del clúster. Si el inicio de sesión es exitoso, se muestra el mensaje "Login Succeeded".
- Etiquete la imagen de ejemplo hpa.

```
docker tag {Image name 1:Tag 1}/{Image repository address}/{Organization name}/  

    {Image name 2:Tag 2}
```

- *{Image name 1:Tag 1}*: nombre y etiqueta de la imagen local que se va a cargar.
- *{Image repository address}*: el nombre de dominio al final del comando de inicio de sesión en **login command**. Se puede obtener en la consola de SWR.
- *{Organization name}*: nombre de la **organización creada**.
- *{Image name 2:Tag 2}*: nombre de imagen y etiqueta deseados que se mostrarán en la consola de SWR.

A continuación, se presenta un ejemplo:

```
docker tag hpa-example:latest swr.ap-southeast-1.myhuaweicloud.com/cloud-develop/hpa-example:latest
```

- Empuje la imagen al repositorio de imágenes.

```
docker push {Image repository address}/{Organization name}/{Image name 2:Tag 2}
```

A continuación, se presenta un ejemplo:

```
docker push swr.ap-southeast-1.myhuaweicloud.com/cloud-develop/hpa-example:latest
```

La siguiente información será devuelta tras un empuje exitoso:

```
6d6b9812c8ae: Pushed  

...  

fe4c16cbf7a4: Pushed  

latest: digest: sha256:eb7e3bbd*** size: **
```

Para ver la imagen enviada, vaya a la consola de SWR y actualice la página **My Images**.

----Fin

## Creación de un grupo de nodos y una política de escalado de nodos

**Paso 1** Inicie sesión en la consola de CCE, acceda al clúster creado, haga clic en **Nodes** a la izquierda, haga clic en la ficha **Node Pools** y haga clic en **Create Node Pool** en la esquina superior derecha.

**Paso 2** Configure el grupo de nodos.

- **Nodes**: Establezca el valor de **1** para indicar que se crea un nodo de forma predeterminada cuando se crea un grupo de nodos.
- **Specifications**: 2 vCPU | 4 GiB

Conservar los valores predeterminados para otros parámetros. Para obtener más información, consulte [Creación de un grupo de nodos](#).

**Paso 3** Busque la fila que contiene el grupo de nodos recién creado y haga clic en **Auto Scaling** en la esquina superior derecha. Para obtener más información, consulte [Creación de una política de escala de nodos](#).

Si el complemento de CCE Cluster Autoscaler no está instalado en el clúster, instálelo primero. Para obtener más información, consulte [CCE Cluster Autoscaler](#).

- **Automatic expansión:** Si esta función está habilitada, los nodos de un grupo de nodos se agregarán automáticamente en función de la carga del clúster.
- **Customized Rule:** Haga clic en **Customized Rule**. En el cuadro de diálogo que se muestra, configure los parámetros. Si la tasa de asignación de CPU es mayor que 70%, se agrega un nodo a cada grupo de nodos asociado. Una política de ajuste de nodos debe estar asociada a un grupo de nodos. Se pueden asociar varios grupos de nodos. Cuando necesite escalar nodos, el nodo con las especificaciones adecuadas se agregará o reducirá del grupo de nodos según el principio de desperdicio mínimo.
- **Automatic reducción:** Si esta función está habilitada, los nodos de un grupo de nodos se eliminarán automáticamente en función de la carga del clúster. Por ejemplo, activar la reducción cuando la utilización de recursos de nodo es inferior al 50%.
- **AS Configuration:** Modifique el rango de cantidades de nodo. Durante el escalado automático, el número de nodos en un grupo de nodos está siempre dentro del rango de cantidades configurado.
- **AS Object:** Habilita el escalado automático para las especificaciones de nodos en un grupo de nodos.

**Paso 4** Haga clic en **OK**.

---Fin

## Creación de una carga de trabajo

Utilice la imagen `hpa-example` para crear un Deployment con una réplica. La ruta de la imagen está relacionada con la organización cargada en el repositorio de SWR y necesita ser reemplazada por el valor real.

```
kind: Deployment
apiVersion: apps/v1
metadata:
  name: hpa-example
spec:
  replicas: 1
  selector:
    matchLabels:
      app: hpa-example
  template:
    metadata:
      labels:
        app: hpa-example
    spec:
      containers:
        - name: container-1
          image: 'hpa-example:latest' # Replace it with the address of the image
you uploaded to SWR.
      resources:
        limits:
          # The value of limits must be the same as that
of requests to prevent flapping during scaling.
          cpu: 500m
          memory: 200Mi
        requests:
          cpu: 500m
          memory: 200Mi
```

```
imagePullSecrets:
- name: default-secret
```

A continuación, cree un Service de NodePort para la carga de trabajo de modo que se pueda acceder a la carga de trabajo desde redes externas.

### NOTA

Para permitir el acceso externo a los Services de NodePort, asigne una EIP para el nodo del clúster. Después de la asignación, sincronice los datos del nodo. Para obtener más información, consulte [Sincronización de datos con servidores en la nube](#). Si el nodo ya está enlazado con una EIP, no es necesario crear uno.

Alternativamente, puede crear un Service con un balanceador de carga de ELB para el acceso externo. Para obtener más información, consulte [Uso de kubectl para crear un Service \(Creación automática de un balanceador de carga compartido\)](#).

```
kind: Service
apiVersion: v1
metadata:
  name: hpa-example
spec:
  ports:
  - name: cce-service-0
    protocol: TCP
    port: 80
    targetPort: 80
    nodePort: 31144
  selector:
    app: hpa-example
  type: NodePort
```

## Creación de una política de HPA

Cree una política de HPA. Como se muestra a continuación, la política está asociada con la carga de trabajo de ejemplo hpa y el uso de CPU de destino es del 50%.

Hay otras dos anotaciones. Una anotación define los umbrales de la CPU, indicando que el ajuste no se realiza cuando el uso de la CPU está entre el 30% y el 70% para evitar el impacto causado por una ligera fluctuación. La otra es la ventana de tiempo de ajuste, que indica que una vez que la política se ejecuta correctamente, una operación de ajuste no se activará de nuevo en este intervalo de enfriamiento para evitar el impacto causado por la fluctuación a corto plazo.

```
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: hpa-policy
  annotations:
    extendedhpa.metrics:
' [{"type": "Resource", "name": "cpu", "targetType": "Utilization", "targetRange": {"low": "30", "high": "70"}} ]'
    extendedhpa.option: '{"downscaleWindow": "5m", "upscaleWindow": "3m"}'
spec:
  scaleTargetRef:
    kind: Deployment
    name: hpa-example
    apiVersion: apps/v1
  minReplicas: 1
  maxReplicas: 100
  metrics:
  - type: Resource
    resource:
      name: cpu
      target:
        type: Utilization
        averageUtilization: 50
```

Configure los parámetros de la siguiente manera si está utilizando la consola.

Pod Range:  ~  When a policy is triggered, the workload pods are scaled within this range.

Cooldown Period: For scale-down  minutes | For scale-up  minutes  
After a policy is successfully triggered, scale-down or scale-up will not triggered again within this cooldown period.

Rules

Metric	Expected Value	Threshold	Operation
CPU usage	50 %	Scale down 30 %   Scale up 70 %	Delete

[Add Rule](#)

## Observación del proceso de escalado automático

**Paso 1** Compruebe el estado del nodo del clúster. En el siguiente ejemplo, hay dos nodos.

```
# kubectl get node
NAME                STATUS    ROLES    AGE    VERSION
192.168.0.183      Ready    <none>   2m20s v1.17.9-r0-CCE21.1.1.3.B001-17.36.8
192.168.0.26       Ready    <none>   55m    v1.17.9-r0-CCE21.1.1.3.B001-17.36.8
```

Verifique la política de HPA. El uso de CPU de la carga de trabajo de destino es 0%.

```
# kubectl get hpa hpa-policy
NAME           REFERENCE                TARGETS  MINPODS  MAXPODS  REPLICAS  AGE
hpa-policy    Deployment/hpa-example    0%/50%   1         100      1          4m
```

**Paso 2** Ejecute el siguiente comando para acceder a la carga de trabajo. En el siguiente comando, {ip:port} indica la dirección de acceso de la carga de trabajo, que se puede consultar en la página de detalles de la carga de trabajo.

```
while true;do wget -q -O- http://{ip:port}; done
```

### 📖 NOTA

Si no se muestra ninguna EIP, al nodo del clúster no se le ha asignado ninguna EIP. Asigne una, vincúlela al nodo y sincronice los datos del nodo. Para obtener más información, consulte [Sincronización de datos con servidores en la nube](#).

Observe el proceso de ajuste de la carga de trabajo.

```
# kubectl get hpa hpa-policy --watch
NAME           REFERENCE                TARGETS  MINPODS  MAXPODS  REPLICAS  AGE
hpa-policy    Deployment/hpa-example    0%/50%   1         100      1          4m
hpa-policy    Deployment/hpa-example    190%/50% 1         100      1          4m23s
hpa-policy    Deployment/hpa-example    190%/50% 1         100      4          4m31s
hpa-policy    Deployment/hpa-example    200%/50% 1         100      4          5m16s
hpa-policy    Deployment/hpa-example    200%/50% 1         100      4          6m16s
hpa-policy    Deployment/hpa-example    85%/50%  1         100      4          7m16s
hpa-policy    Deployment/hpa-example    81%/50%  1         100      4          8m16s
hpa-policy    Deployment/hpa-example    81%/50%  1         100      7          8m31s
hpa-policy    Deployment/hpa-example    57%/50%  1         100      7          9m16s
hpa-policy    Deployment/hpa-example    51%/50%  1         100      7          10m
```

```
hpa-policy    Deployment/hpa-example    58%/50%    1    100    7
11m
```

Puede ver que el uso de la CPU de la carga de trabajo es del 190% a 4m23s, lo que excede el valor objetivo. En este caso, se activa el ajuste para expandir la carga de trabajo a cuatro réplicas/pods. En los siguientes minutos, el uso de la CPU no disminuye hasta 7m16s. Esto se debe a que es posible que los nuevos pods no se creen correctamente. La posible causa es que los recursos son insuficientes y los pods están en estado Pending. Durante este período, se agregan nodos.

A 7m16s, el uso de la CPU disminuye, lo que indica que los pods se crean con éxito y comienzan a soportar tráfico. El uso de CPU disminuye a 81% a 8m, aún mayor que el valor objetivo (50%) y el umbral alto (70%). Por lo tanto, se agregan 7 pods a 9m16s, y el uso de CPU disminuye a 51%, que está dentro del rango de 30% a 70%. A partir de entonces, el número de pods sigue siendo 7.

En el siguiente resultado, puede ver el proceso de ajuste de la carga de trabajo y el momento en que la política de HPA entra en vigor.

```
# kubectl describe deploy hpa-example
...
Events:
  Type      Reason          Age   From          Message
  ----      -
  Normal    ScalingReplicaSet 25m   deployment-controller Scaled up replica set hpa-example-79dd795485 to 1
  Normal    ScalingReplicaSet 20m   deployment-controller Scaled up replica set hpa-example-79dd795485 to 4
  Normal    ScalingReplicaSet 16m   deployment-controller Scaled up replica set hpa-example-79dd795485 to 7
# kubectl describe hpa hpa-policy
...
Events:
  Type      Reason          Age   From          Message
  ----      -
  Normal    SuccessfulRescale 20m   horizontal-pod-autoscaler New size: 4; reason: cpu resource utilization (percentage of request) above target
  Normal    SuccessfulRescale 16m   horizontal-pod-autoscaler New size: 7; reason: cpu resource utilization (percentage of request) above target
```

Compruebe el número de nodos. El siguiente resultado muestra que se agregan dos nodos.

```
# kubectl get node
NAME           STATUS    ROLES    AGE   VERSION
192.168.0.120  Ready    <none>   3m5s v1.17.9-r0-CCE21.1.1.3.B001-17.36.8
192.168.0.136  Ready    <none>   6m58s v1.17.9-r0-CCE21.1.1.3.B001-17.36.8
192.168.0.183  Ready    <none>   18m   v1.17.9-r0-CCE21.1.1.3.B001-17.36.8
192.168.0.26   Ready    <none>   71m   v1.17.9-r0-CCE21.1.1.3.B001-17.36.8
```

También puede ver el historial de ajuste en la consola. Por ejemplo, la política de CA se ejecuta una vez cuando la tasa de asignación de CPU en el clúster es mayor que 70%, y el número de nodos en el grupo de nodos aumenta de 2 a 3. El nuevo nodo se agrega automáticamente por el autoscaler basado en el estado pendiente de los pods en la fase inicial de HPA.

El proceso de ajuste de nodos es el siguiente:

1. Después de que el número de pods cambia a 4, los pods están en estado Pending debido a recursos insuficientes. Como resultado, se activa la política de expansión predeterminada del complemento del autoscaler, y el número de nodos se incrementa en uno.
2. El segundo nodo expansión se activa porque la tasa de asignación de CPU en el clúster es superior al 70%. Como resultado, el número de nodos aumenta en uno, que se registra

en el historial de ajuste en la consola. El escalado basado en la tasa de asignación garantiza que el clúster tenga suficientes recursos.

**Paso 3** Deje de acceder a la carga de trabajo y compruebe el número de pods.

```
# kubectl get hpa hpa-policy --watch
```

NAME	REFERENCE	TARGETS	MINPODS	MAXPODS	REPLICAS
AGE					
hpa-policy	Deployment/hpa-example	50%/50%	1	100	7
12m					
hpa-policy	Deployment/hpa-example	21%/50%	1	100	7
13m					
hpa-policy	Deployment/hpa-example	0%/50%	1	100	7
14m					
hpa-policy	Deployment/hpa-example	0%/50%	1	100	7
18m					
hpa-policy	Deployment/hpa-example	0%/50%	1	100	3
18m					
hpa-policy	Deployment/hpa-example	0%/50%	1	100	3
19m					
hpa-policy	Deployment/hpa-example	0%/50%	1	100	3
19m					
hpa-policy	Deployment/hpa-example	0%/50%	1	100	3
19m					
hpa-policy	Deployment/hpa-example	0%/50%	1	100	3
23m					
hpa-policy	Deployment/hpa-example	0%/50%	1	100	3
23m					
hpa-policy	Deployment/hpa-example	0%/50%	1	100	1
23m					

Puede ver que el uso de la CPU es del 21% a 13m. El número de pods se reduce a 3 a 18m, y luego se reduce a 1 a 23m.

En el siguiente resultado, puede ver el proceso de ajuste de la carga de trabajo y el momento en que la política de HPA entra en vigor.

```
# kubectl describe deploy hpa-example
...
Events:
  Type      Reason          Age   From                      Message
  ----      -
  Normal    ScalingReplicaSet 25m   deployment-controller     Scaled up replica set hpa-example-79dd795485 to 1
  Normal    ScalingReplicaSet 20m   deployment-controller     Scaled up replica set hpa-example-79dd795485 to 4
  Normal    ScalingReplicaSet 16m   deployment-controller     Scaled up replica set hpa-example-79dd795485 to 7
  Normal    ScalingReplicaSet 6m28s deployment-controller     Scaled down replica set hpa-example-79dd795485 to 3
  Normal    ScalingReplicaSet 72s   deployment-controller     Scaled down replica set hpa-example-79dd795485 to 1
# kubectl describe hpa hpa-policy
...
Events:
  Type      Reason          Age   From                      Message
  ----      -
  Normal    SuccessfulRescale 20m   horizontal-pod-autoscaler New size: 4; reason: cpu resource utilization (percentage of request) above target
  Normal    SuccessfulRescale 16m   horizontal-pod-autoscaler New size: 7; reason: cpu resource utilization (percentage of request) above target
  Normal    SuccessfulRescale 6m45s horizontal-pod-autoscaler New size: 3; reason: All metrics below target
  Normal    SuccessfulRescale 90s   horizontal-pod-autoscaler New size: 1; reason: All metrics below target
```

También puede ver el historial de ejecución de políticas de HPA en la consola. Espere hasta que se reduzca el nodo.

La razón por la que los otros dos nodos en el grupo de nodos no se reducen es que ambos tienen pods en el espacio de nombres del sistema kube (y estos pods no son creados por DaemonSets). Para obtener más información, consulte [Mecanismos de escala de nodos](#).

----Fin

## Resumen

El uso de HPA y CA puede implementar fácilmente el ajuste automático en la mayoría de los escenarios. Además, el proceso de ajuste de nodos y pods se puede observar fácilmente.

# 14 Complementos

## 14.1 Descripción general

CCE proporciona múltiples tipos de complementos para ampliar las funciones de clúster y cumplir con los requisitos de características. Puede instalar complementos según sea necesario.

### AVISO

CCE utiliza plantillas Helm para desplegar complementos. Para modificar o actualizar un complemento, realice operaciones en la página **Add-ons** o use las API abiertas. No modifique directamente los recursos relacionados con los complementos en segundo plano. De lo contrario, pueden producirse excepciones de complementos u otros problemas inesperados.

**Tabla 14-1** Lista de complementos

Nombre del complemento	Presentación
<b>coredns</b> (complemento de recursos del sistema, obligatorio)	El complemento coredns es un servidor DNS que proporciona servicios de resolución de nombres de dominio para clústeres de Kubernetes. Complementos de coredns cadenas para proporcionar características adicionales.
<b>storage-driver</b> (complemento de recursos del sistema, descartado)	storage-driver es un controlador de FlexVolume utilizado para admitir servicios de almacenamiento IaaS como EVS, SFS y OBS.

Nombre del complemento	Presentación
<b>everest</b> (complemento de recursos del sistema, obligatorio)	Everest es un sistema de almacenamiento de contenedor nativo de la nube. Según Container Storage Interface (CSI), los clústeres de Kubernetes v1.15.6 o posterior obtienen acceso a los servicios de almacenamiento en la nube.
<b>npd</b>	node-problem-detector (npd para abreviar) es un complemento que monitorea eventos anormales de nodos de clúster y se conecta a una plataforma de monitoreo de terceros. Es un demonio que se ejecuta en cada nodo. Recopila problemas de nodos de diferentes demonios y los informa al servidor de API. El complemento npd puede ejecutarse como un demonio o un DaemonSet.
<b>dashboard</b>	Panel de Kubernetes es una interfaz de usuario de propósito general basada en web para clústeres de Kubernetes e integra todos los comandos que se pueden usar en la CLI. Permite a los usuarios gestionar aplicaciones que se ejecutan en un clúster y solucionar fallas, así como gestionar el clúster en sí.
<b>autoscaler</b>	El complemento del autoscaler cambia el tamaño de un clúster según el estado de programación de pods y el uso de recursos.
<b>metrics-server</b>	metrics-server es un agregador para monitorear los datos de los recursos del clúster central.
<b>cce-hpa-controller</b>	cce-hpa-controller es un complemento desarrollado por CCE, que se puede utilizar para escalar de manera flexible las Deployments basadas en métricas como el uso de la CPU y el uso de la memoria.
<b>prometheus</b>	Prometheus es un marco de monitoreo y alerta de sistema de código abierto. CCE le permite instalar rápidamente Prometheus como complemento.
<b>web-terminal</b>	web-terminal es un complemento que permite usar kubectl en una interfaz de usuario web. Se puede conectar a Linux usando WebSocket con un navegador y proporciona las API para la integración en sistemas independientes. Se puede utilizar directamente como servicio para obtener información con la base de datos de gestión de configuración (CMDB) e iniciar sesión en el servidor.
<b>gpu-device-plugin</b> (formerly gpu-beta)	gpu-device-plugin es un complemento de gestión de dispositivos que admite GPU de contenedores. Solo es compatible con los controladores de NVIDIA.
<b>huawei-npu</b>	Huawei-npu es un complemento de gestión para dispositivos de Huawei NPU en contenedores.

Nombre del complemento	Presentación
<b>volcano</b>	Volcano proporciona capacidades informáticas de alto rendimiento de propósito general, como la programación de trabajos, la gestión de chips heterogéneos y la gestión de ejecución de trabajos, sirviendo a los usuarios finales con marcos informáticos para diferentes industrias, como la IA, el big data, la secuenciación de genes y el renderizado.
<b>nginx-ingress</b>	nginx-ingress proporciona funciones de reenvío de la capa de aplicación, como hosts virtuales, equilibrio de carga, proxy SSL y enrutamiento HTTP, para Services a los que se puede acceder directamente fuera de un clúster.
<b>dew-provider</b>	El complemento de dew-provider se utiliza para interconectar con <b>Data Encryption Workshop (DEW)</b> , que le permite montar secretos almacenados fuera de un clúster (es decir, DEW para almacenar información confidencial) en los pods. De esta manera, la información sensible puede desacoplarse del entorno de agrupamiento, evitando la fuga de información causada por la configuración de codificación dura del programa o de texto plano.
<b>dolphin</b>	<p>dolphin es un complemento de monitoreo a la red de pod, que, en la versión actual, se puede utilizar para recopilar estadísticas sobre el tráfico de red pública de contenedores Kata en clústeres de CCE Turbo y los contenedores comunes que utilizan containerd como tiempo de ejecución.</p> <p>Este complemento recopila cuántos paquetes de IPv4 y bytes se reciben y envían (incluidos los enviados a la red pública).</p> <p>PodSelectors se puede usar para seleccionar backends de monitoreo para admitir múltiples tareas de monitoreo y métricas de monitoreo opcionales. También puede obtener información sobre la etiqueta de los pods. La información de seguimiento se ha adaptado al formato de Prometheus. Puede invocar a la API de Prometheus para ver los datos de monitoreo.</p>
<b>node-local-dns</b>	NodeLocal DNSCache mejora el rendimiento de DNS del clúster al ejecutar proxys de caché de DNS como DaemonSets en los nodos del clúster.

## Operaciones relacionadas

Puede realizar las operaciones descritas en [Tabla 14-2](#) en la página **Add-ons**.

**Tabla 14-2** Operaciones relacionadas

Opera ción	Descripción	Procedimiento
Instalar	Instalar un complemento especificado.	<ol style="list-style-type: none"> <li>1. Inicie sesión en la consola de CCE y haga clic en el nombre del clúster para acceder a la consola del clúster. En el panel de navegación, elija <b>Add-ons</b>.</li> <li>2. Haga clic en <b>Install</b> en el complemento de destino. Cada complemento tiene diferentes parámetros de configuración. Para más detalles, consulte el capítulo correspondiente.</li> <li>3. Haga clic en <b>OK</b>.</li> </ol>
Actualizar	Actualizar un complemento a la nueva versión.	<ol style="list-style-type: none"> <li>1. Inicie sesión en la consola de CCE y haga clic en el nombre del clúster para acceder a la consola del clúster. En el panel de navegación, elija <b>Add-ons</b>.</li> <li>2. Si se puede actualizar un complemento, aparece el botón <b>Upgrade</b> debajo de él. Haga clic en <b>Upgrade</b>. Cada complemento tiene diferentes parámetros de configuración. Para más detalles, consulte el capítulo correspondiente.</li> <li>3. Haga clic en <b>OK</b>.</li> </ol>
Editar	Editar parámetros adicionales.	<ol style="list-style-type: none"> <li>1. Inicie sesión en la consola de CCE y haga clic en el nombre del clúster para acceder a la consola del clúster. En el panel de navegación, elija <b>Add-ons</b>.</li> <li>2. Haga clic en <b>Edit</b> en el complemento de destino. Cada complemento tiene diferentes parámetros de configuración. Para más detalles, consulte el capítulo correspondiente.</li> <li>3. Haga clic en <b>OK</b>.</li> </ol>
Desinstalar	Desinstalar un complemento del clúster.	<ol style="list-style-type: none"> <li>1. Inicie sesión en la consola de CCE y haga clic en el nombre del clúster para acceder a la consola del clúster. En el panel de navegación, elija <b>Add-ons</b>.</li> <li>2. Haga clic en <b>Uninstall</b> en el complemento de destino.</li> <li>3. En el cuadro de diálogo que se muestra, haga clic en <b>Yes</b>. Esta operación no se puede deshacer.</li> </ol>

## 14.2 coredns (complemento de recursos del sistema, obligatorio)

### Presentación

El complemento coredns es un servidor DNS que proporciona servicios de resolución de nombres de dominio para clústeres de Kubernetes. Complementos de coredns cadenas para proporcionar características adicionales.

coredns es un software de código abierto y ha sido parte de CNCF. Proporciona un medio para que los servicios en la nube se descubran entre sí en despliegues nativos en la nube. Cada uno de los complementos encadenados por coredns proporciona una función de DNS particular. Puede integrar coredns con solo los complementos que necesite para que sea rápido, eficiente y flexible. Cuando se utiliza en un clúster de Kubernetes, coredns puede detectar automáticamente los servicios del clúster y proporcionar la resolución de nombres de dominio para estos servicios. Al trabajar con el servidor de DNS, coredns puede resolver nombres de dominio externos para cargas de trabajo en un clúster.

**coredns es un complemento de recursos del sistema. Se instala de forma predeterminada cuando se crea un clúster de Kubernetes v1.11 o posterior.**

Kubernetes v1.11 y posteriores respaldan a CoreDNS como el DNS predeterminado oficial para todos los clústeres en el futuro.

Sitio web oficial de CoreDNS: <https://coredns.io/>

Comunidad de código abierto: <https://github.com/coredns/coredns>

#### NOTA

Para obtener más información, véase [DNS](#).

### Notas y restricciones

Cuando coredns se está ejecutando correctamente o se está actualizando, asegúrese de que el número de nodos disponibles es mayor o igual que el número de instancias de coredns y todas las instancias de coredns se están ejecutando. De lo contrario, la actualización fallará.

### Instalación del complemento

Este complemento se ha instalado de forma predeterminada. Si se desinstala por alguna razón, puede volver a instalarlo realizando los siguientes pasos:

- Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster. Elija **Add-ons** en el panel de navegación, localice **coredns** a la derecha y haga clic en **Install**.
- Paso 2** En la página **Install Add-on**, seleccione las especificaciones del complemento y establezca los parámetros relacionados.

**Tabla 14-3** parámetros de complemento de coredns

Parámetro	Descripción
Add-on Specifications	<p>Capacidad de resolución de nombres de dominio simultáneos. Seleccione las especificaciones adicionales que mejor se adapten a sus necesidades.</p> <p>Si selecciona <b>Custom qps</b>, la resolución de nombres de dominio de QPS proporcionada por CoreDNS se correlaciona positivamente con el consumo de CPU. Ajuste el número de pods y las cuotas de memoria y CPU de contenedor según sea necesario.</p>
Pods	Número de pods que se crearán para que coincidan con las especificaciones del complemento seleccionado.
Containers	Cuotas de CPU y memoria del contenedor permitidas para las especificaciones de complemento seleccionadas.

Parámetro	Descripción
Parameters	<ul style="list-style-type: none"> <li>● <b>parameterSyncStrategy</b>: indica si se debe configurar la comprobación de consistencia cuando se actualiza un complemento.                             <ul style="list-style-type: none"> <li>– <b>ensureConsistent</b>: indica que la comprobación de consistencia de la configuración está activada. Si la configuración registrada en el clúster es incompatible con la configuración real, el complemento no se puede actualizar.</li> <li>– <b>force</b>: indica que la comprobación de consistencia de la configuración se ignora durante una actualización. Asegúrese de que la configuración efectiva actual es la misma que la configuración original. Después de actualizar el complemento, restaure el valor de <b>parameterSyncStrategy</b> a <b>ensureConsistent</b> y vuelva a activar la comprobación de consistencia de configuración.</li> <li>– <b>inherit</b>: indica que las configuraciones diferenciadas se heredan automáticamente durante una actualización. Después de actualizar el complemento, restaure el valor de <b>parameterSyncStrategy</b> a <b>ensureConsistent</b> y vuelva a activar la comprobación de consistencia de configuración.</li> </ul> </li> <li>● <b>stub_domains</b>: Servidor de nombres de dominio para un nombre de dominio definido por el usuario. El formato es un par clave-valor. La clave es un sufijo de nombre de dominio de DNS y el valor es una o más direcciones IP de DNS.</li> <li>● <b>upstream_nameservers</b>: La dirección IP del servidor de DNS ascendente.</li> <li>● <b>servers</b>: La configuración de servidores está disponible desde coredns 1.23.1. Puede personalizar la configuración de los servidores. Para obtener más información, consulte <a href="https://coredns.io/manual/plugins/">dns-custom-nameservers</a>. <b>plugins</b> indica la configuración de cada componente en coredns (<a href="https://coredns.io/manual/plugins/">https://coredns.io/manual/plugins/</a>). Se recomienda conservar las configuraciones predeterminadas en escenarios comunes para evitar que CoreDNS no esté disponible debido a errores de configuración. Cada componente del complemento contiene <b>name</b>, <b>parameters</b> (opcional) y <b>configBlock</b> (opcional). El formato del Corefile generado es el siguiente:                             <pre>\$name \$parameters { \$configBlock }</pre> <p><b>Tabla 14-4</b> describe los complementos comunes.</p> <p>Por ejemplo:</p> <pre>{   "servers": [     {       "plugins": [         {           "name": "bind",           "parameters": "{\$POD_IP}"         },         {</pre> </li> </ul>

Parámetro	Descripción
	<pre>         "name": "cache",         "parameters": 30       },       {         "name": "errors"       },       {         "name": "health",         "parameters": "\${POD_IP}:8080"       },       {         "configBlock": "pods insecure \nfallthrough in-addr.arpa ip6.arpa",         "name": "kubernetes",         "parameters": "cluster.local in- addr.arpa ip6.arpa"       },       {         "name": "loadbalance",         "parameters": "round_robin"       },       {         "name": "prometheus",         "parameters": "\${POD_IP}:9153"       },       {         "configBlock": "policy random",         "name": "forward",         "parameters": ". /etc/resolv.conf"       },       {         "name": "reload"       },       {         "name": "log"       }     ],     "port": 5353,     "zones": [       {         "zone": "."       }     ]   },   "stub_domains": {     "acme.local": [       "1.2.3.4",       "6.7.8.9"     ]   },   "upstream_nameservers": ["8.8.8.8", "8.8.4.4"] } </pre>

**Tabla 14-4** Configuración predeterminada del complemento de la zona activa de coredns

Nombre del complemento	Descripción
bind	Dirección IP del host escuchada por coredns. Se recomienda conservar el valor por defecto <b>{SPOD_IP}</b> .

Nombre del complemento	Descripción
cache	La caché de DNS está habilitada.
errors	Los errores se registran en stdout.
health	Configuración de comprobación de estado. La dirección IP de escucha actual es <code>{\$POD_IP}:8080</code> . Conserve el valor predeterminado. De lo contrario, la comprobación de estado de <code>coredns</code> falla y <code>coredns</code> se reinicia repetidamente.
kubernetes	Complemento de CoreDNS Kubernetes, que proporciona la capacidad de análisis de servicios en un clúster.
loadbalance	Balancedor de carga de DNS de asignación cíclica que aleatoriza el orden de los registros A, AAAA y MX en la respuesta.
prometheus	Puerto para obtener métricas de <code>coredns</code> . La dirección IP de escucha de zona predeterminada es <code>{\$POD_IP}:9153</code> . Conserve el valor predeterminado. De lo contrario, CloudScope no pueden recopilar métricas de <code>coredns</code> .
forward	Las consultas que no estén dentro del dominio de clúster de Kubernetes se reenviarán a solucionador predefinidos ( <code>/etc/resolv.conf</code> ).
reload	El Corefile cambiado se puede recargar automáticamente. Después de editar el ConfigMap espere dos minutos para que la modificación surta efecto.

**Paso 3** Una vez completadas las configuraciones anteriores, haga clic en **Install**.

---Fin

## ¿Cómo funciona la resolución de nombres de dominio en Kubernetes?

Las políticas de DNS se pueden establecer en función de cada pod. Actualmente, Kubernetes admite cuatro tipos de políticas de DNS: **Default**, **ClusterFirst**, **ClusterFirstWithHostNet** y **None**. Para obtener más información, véase <https://kubernetes.io/docs/concepts/services-networking/dns-pod-service/>. Estas políticas se especifican en el campo `dnsPolicy` del pod específico.

- **Default:** Los pods heredan la configuración de resolución de nombres del nodo en el que se ejecutan los pods. El servidor de DNS ascendente personalizado y el dominio stub no se pueden usar junto con esta política.
- **ClusterFirst:** Cualquier consulta de DNS que no coincida con el sufijo de dominio de clúster configurado, como `www.kubernetes.io`, se reenvía al servidor de nombres de flujo ascendente heredado del nodo. Los administradores de clústeres pueden tener otros dominios stub y servidores de DNS ascendentes configurados.
- **ClusterFirstWithHostNet:** Para los pods que se ejecutan con `hostNetwork`, establezca su política de DNS **ClusterFirstWithHostNet**.

- **None:** Permite que un pod ignore la configuración de DNS del entorno de Kubernetes. Se supone que todas las configuraciones de DNS se proporcionan usando el campo **dnsPolicy** en el pod específico.

**NOTA**

- Los clústeres de Kubernetes v1.10 y posteriores admiten **Default**, **ClusterFirst**, **ClusterFirstWithHostNet** y **None**. Los clústeres anteriores a Kubernetes v1.10 solo admiten **Default**, **ClusterFirst** y **ClusterFirstWithHostNet**.
- **Default** no es la política de DNS predeterminada. Si **dnsPolicy** no se especifica explícitamente, se utiliza **ClusterFirst**.

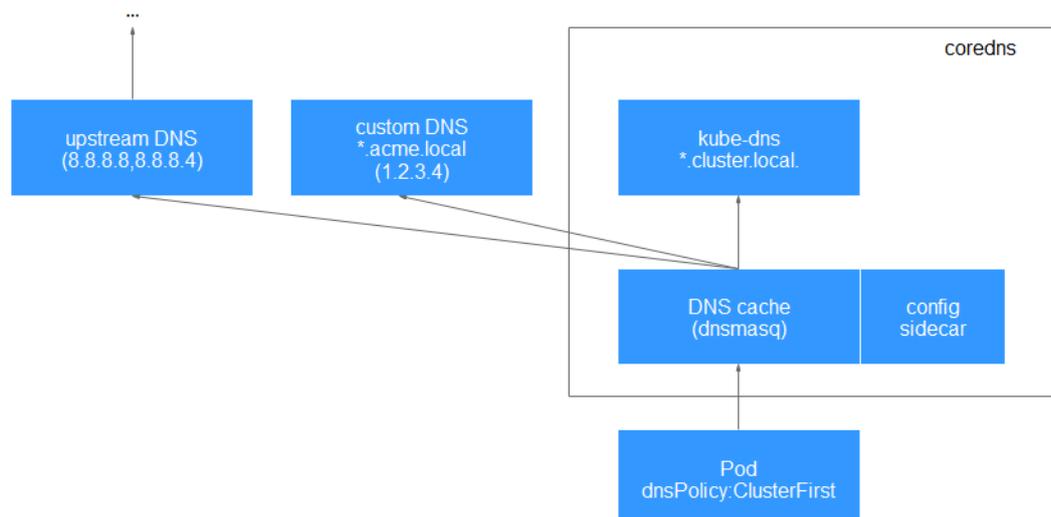
**Routing**

**Without stub domain configurations:** cualquier consulta que no coincida con el sufijo de dominio de clúster configurado, como **www.kubernetes.io**, se reenvía al servidor de DNS ascendente heredado del nodo.

**With stub domain configurations:** Si se configuran dominios stub y servidores de DNS ascendentes, las consultas de DNS se enrutan según el siguiente flujo:

1. La consulta se envía primero a la capa de caché de DNS en coredns.
2. Desde la capa de almacenamiento en caché, se examina el sufijo de la solicitud y luego la solicitud se reenvía al DNS correspondiente:
  - Nombres con el sufijo de clúster, por ejemplo, **.cluster.local**: La solicitud se envía a coredns.
  - Nombres con el sufijo de dominio stub, por ejemplo, **.acme.local**: La solicitud se envía al solucionador de DNS personalizado configurado que escucha, por ejemplo, en 1.2.3.4.
  - Nombres que no coinciden con el sufijo (por ejemplo, **widget.com**): la solicitud se reenvía al DNS ascendente.

**Figura 14-1** Enrutamiento



**Personalización de las especificaciones de CoreDNS**

En la consola, el complemento de coredns solo se puede configurar con las especificaciones preestablecidas, que pueden satisfacer la mayoría de los requisitos de servicio. En algunos

escenarios en los que hay requisitos sobre el uso de recursos de CoreDNS, es posible que deba personalizar las especificaciones del complemento.

A continuación se describe cómo personalizar los parámetros de CoreDNS, incluidas las réplicas, las CPU y la memoria.

### AVISO

- La modificación incorrecta en la configuración de CoreDNS puede provocar errores de resolución de nombres de dominio en el clúster. Realice pruebas antes y después de la modificación.
- Al modificar el número de réplicas de CoreDNS, CPUs y tamaño de memoria, se cambiará la capacidad de análisis de CoreDNS. Por lo tanto, evalúe el impacto antes de la operación.
- De forma predeterminada, podAntiAffinity (pod anti-afinidad) está configurado para el complemento de coredns. Si un nodo ya tiene un pod de CoreDNS, no se puede agregar un pod nuevo. Es decir, solo un pod de CoreDNS puede ejecutarse en un nodo. Si el número de réplicas de CoreDNS configuradas es mayor que el número de nodos de clúster, el exceso de pods no se puede programar. Por lo tanto, mantenga el número de réplicas menor o igual que el número de nodos.

**Paso 1** Invoque a la [API de consulta de complementos](#) para consultar el ID del complemento de coredns instalado.

GET https://{cluster\_id}.{endpoint}/api/v3/addons?cluster\_id={cluster\_id}

Para obtener más información sobre cómo invocar a una API, consulte [Invocación de las API](#). {cluster\_id} indica el ID del clúster, que se puede ver en la página de detalles del clúster en la consola de CCE. {endpoint} indica la dirección de acceso de CCE, que se puede obtener de [Regiones y puntos de conexión](#). Por ejemplo, el `cce.east-3.myhuaweicloud.com` indica la región de Shanghai.

Vea el complemento `uid` y `spec` de coredns en la respuesta.

```
{
  "kind": "Addon",
  "apiVersion": "v3",
  "items": [
    {
      "kind": "Addon",
      "apiVersion": "v3",
      "metadata": {
        "uid": "2083381d-46ae-11ec-91a8-0255ac1000c5",
        "name": "coredns",
        "creationTimestamp": "2021-11-16T07:23:42Z",
        "updateTimestamp": "2021-11-16T07:27:35Z"
      },
      "spec": {
        "clusterID": "15d748b4-3de1-11ec-9199-0255ac1000c9",
        "version": "1.17.9",
        "addonTemplateName": "coredns",
        "addonTemplateType": "helm",
        ...
      }
    }
  ]
}
```

**Paso 2** Invoque a la [API de configuración](#) para modificar las especificaciones del complemento.

PUT https://{cluster\_id}.{endpoint}/api/v3/addons/{uid}

**{cluster\_id}** indica el ID del clúster. **{endpoint}** indica la dirección para acceder a CCE, que se puede obtener de [Regiones y puntos de conexión](#). **{uid}** indica el ID de complemento obtenido en el paso anterior.

A continuación se muestra un ejemplo del cuerpo de la solicitud. **spec** define el contenido consultado en el paso anterior. Modifique **replicas** (número de réplicas) y **resources** (uso de CPU/memoria de un solo pod) según sea necesario. Conserve los valores de **clusterID** y **version**. El valor de **addon.upgrade/type** debe ser **patch**.

```
{
  "metadata": {
    "annotations": {
      "addon.upgrade/type": "patch"
    }
  },
  "spec": {
    "clusterID": "15d748b4-3de1-11ec-9199-0255ac1000c9",
    "version": "1.17.9",
    "addonTemplateName": "coredns",
    "values": {
      "flavor": {
        "replicas": 2,
        "resources": [
          {
            "limitsCpu": "500m",
            "limitsMem": "512Mi",
            "requestsCpu": "500m",
            "requestsMem": "512Mi"
          }
        ]
      }
    }
  }
}
```

---Fin

## Historial de cambios

Tabla 14-5 Versiones de complementos de CCE

Versión del complemento	Versión de clúster admitida	Versión de la comunidad (solo para clústeres de v1.17 y posteriores)
1.25.1	/v1.(19 21 23 25).*/	<b>1.8.4</b>
1.23.3	/v1.(15 17 19 21 23).*/	<b>1.8.4</b>
1.23.2	/v1.(15 17 19 21 23).*/	<b>1.8.4</b>
1.23.1	/v1.(15 17 19 21 23).*/	<b>1.8.4</b>
1.17.15	/v1.(15 17 19 21).*/	<b>1.8.4</b>
1.17.9	/v1.(15 17 19).*/	<b>1.8.4</b>
1.17.7	/v1.(15 17 19).*/	<b>1.8.4</b>
1.17.4	/v1.(17 19).*/	<b>1.6.5</b>

Versión del complemento	Versión de clúster admitida	Versión de la comunidad (solo para clústeres de v1.17 y posteriores)
1.17.3	/v1.17.*/	<b>1.6.5</b>
1.17.1	/v1.17.*/	<b>1.6.5</b>

## 14.3 everest (complemento de recursos del sistema, obligatorio)

### Presentación

Everest es un sistema de almacenamiento de contenedor nativo de la nube. Según Container Storage Interface (CSI), los clústeres de Kubernetes v1.15.6 o posterior obtienen acceso a los servicios de almacenamiento en la nube.

**everest es un complemento de recursos del sistema. Se instala de forma predeterminada cuando se crea un clúster de Kubernetes v1.15 o posterior.**

### Notas y restricciones

- Si el clúster se actualiza de v1.13 a v1.15, el **storage-driver** se sustituye por everest (v1.1.6 o posterior) para el almacenamiento de contenedor. La adquisición no afecta a las funciones de almacenamiento originales.
- En la versión 1.2.0 del complemento más antiguo, se optimiza **key authentication** cuando se utiliza OBS. Después de actualizar el complemento everest desde una versión anterior a 1.2.0, debe reiniciar todas las cargas de trabajo que utilizan OBS en el clúster. De lo contrario, es posible que las cargas de trabajo no puedan usar OBS.
- De forma predeterminada, este complemento se instala en los **clústeres de v1.15 y posteriores**. Para los clústeres de v1.13 y versiones anteriores, el complemento **storage-driver** se instala de forma predeterminada.

### Instalación del complemento

Este complemento se ha instalado de forma predeterminada. Si se desinstala por alguna razón, puede volver a instalarlo realizando los siguientes pasos:

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster. Elija **Add-ons** en el panel de navegación, localice **everest** a la derecha y haga clic en **Install**.

**Paso 2** Seleccione **Standalone, HA o Custom** para **Add-on Specifications**.

El complemento everest contiene los contenedores siguientes. Puede ajustar las especificaciones según sea necesario.

- **everest-csi-controller**: Una carga de trabajo de Deployment. Este contenedor es responsable de crear, eliminar, crear instantáneas, expandir, enlazar y separar volúmenes. Si la versión del clúster es 1.19 o posterior y la versión del complemento es 1.2.x es el pod del componente de controlador siempre csi también tiene un contenedor de everest-

localvolume-manager por defecto. Este contenedor gestiona la creación de grupos de almacenamiento de LVM y PV locales en el nodo.

 **NOTA**

Si selecciona **Custom**, la configuración de memoria recomendada para **everest-csi-controller** es la siguiente:

- Si el número de pods y PVCs es inferior a 2000, establezca el límite superior de memoria en 600 MiB.
  - Si el número de pods y PVCs es inferior a 5000, establezca el límite superior de memoria en 1 GiB.
- **everest-csi-driver**: Una carga de trabajo de DaemonSet. Este contenedor es responsable de montar y desmontar los PV y cambiar el tamaño de los sistemas de archivos. Si la versión del complemento es 1.2.x y la región donde se encuentra el clúster soporta node-attacher, el pod del componente everest-csi-driver también contiene un contenedor everest-node-attacher. Este contenedor es responsable de la conexión distribuida de EVS. Este elemento de configuración está disponible en algunas regiones.

 **NOTA**

Si selecciona **Custom**, el límite de memoria recomendado para **everest-csi-driver** es de 300 MiB o superior. Si el valor es demasiado pequeño, el complemento de contenedor no se puede iniciar y el complemento no está disponible.

### Paso 3 Configure los parámetros relacionados.

En everest 1.2.26 o posterior, se optimiza el rendimiento de la conexión de un gran número de volúmenes de EVS. Se proporcionan los siguientes tres parámetros:

- **csi\_attacher\_worker\_threads**: número de trabajadores que pueden montar simultáneamente los volúmenes de EVS. El valor predeterminado es **60**.
- **csi\_attacher\_detach\_worker\_threads**: número de trabajadores que pueden desmontar simultáneamente los volúmenes de EVS. El valor predeterminado es **60**.
- **volume\_attaching\_flow\_ctrl**: número máximo de volúmenes de EVS que se pueden montar con el complemento más antiguo en un minuto. El valor predeterminado es **0**, lo que indica que el rendimiento de montaje del volumen de EVS está determinado por los recursos de almacenamiento subyacentes.

Los tres parámetros anteriores están asociados entre sí y están limitados por los recursos de almacenamiento subyacentes en la región donde se encuentra el clúster. Si desea montar un gran número de volúmenes (más de 500 volúmenes de EVS por minuto), puede ponerse en contacto con el personal de servicio al cliente y configurar los parámetros bajo su guía para evitar que el complemento más antiguo se ejecute de forma anormal debido a una configuración de parámetros incorrecta.

Otros parámetros

- **cluster\_id**: ID del clúster.
- **default\_vpc\_id**: ID de la VPC a la que pertenece el clúster del almacén de datos.
- **disable\_auto\_mount\_secret**: indica si se puede utilizar la AK/SK predeterminada cuando se monta un bucket de objetos o un sistema de archivos paralelo. El valor predeterminado es **false**.
- **enable\_node\_attacher**: indica si se debe habilitar el adjunto en el agente para procesar el **VolumeAttachment**.
- **flow\_control**: Este parámetro se deja en blanco por defecto.

- **over\_subscription**: ratio de sobrecompromiso del grupo de almacenamiento local (**local\_storage**). El valor predeterminado es **80**. Si el tamaño del grupo de almacenamiento local es de 100 GB, puede asignar más de 180 GB.
- **project\_id**: ID del proyecto al que pertenece el clúster.

**Paso 4** Haga clic en **Install**.

---Fin

## Historial de cambios

**Tabla 14-6** Versiones de complementos de CCE

Versión del complemento	Versión de clúster admitida
2.1.29	/v1.(19 21 23 25).*/
2.1.14	/v1.(19 21 23 25).*/
2.1.13	/v1.(19 21 23 25).*/
2.1.9	/v1.(19 21 23 25).*/
2.0.9	/v1.(19 21 23).*/
1.3.28	/v1.(19 21 23).*/
1.3.22	/v1.(19 21 23).*/
1.3.20	/v1.(19 21 23).*/
1.3.17	/v1.(19 21 23).*/
1.3.8	/v1.23.*/
1.3.6	/v1.23.*/
1.2.55	/v1.(15 17 19 21).*/
1.2.53	/v1.(15 17 19 21).*/
1.2.51	/v1.(15 17 19 21).*/
1.2.44	/v1.(15 17 19 21).*/
1.2.42	/v1.(15 17 19 21).*/
1.2.30	/v1.(15 17 19 21).*/
1.2.28	/v1.(15 17 19 21).*/
1.2.27	/v1.(15 17 19 21).*/
1.2.13	/v1.(15 17 19).*/
1.2.9	/v1.(15 17 19).*/
1.2.5	/v1.(15 17 19).*/

Versión del complemento	Versión de clúster admitida
1.1.12	/v1.(15 17).*/
1.1.11	/v1.(15 17).*/
1.1.8	/v1.(15 17).*/
1.1.7	/v1.(15 17).*/

## 14.4 npd

### Presentación

node-problem-detector (npd para abreviar) es un complemento que monitorea eventos anormales de nodos de clúster y se conecta a una plataforma de monitoreo de terceros. Es un demonio que se ejecuta en cada nodo. Recopila problemas de nodos de diferentes demonios y los informa al servidor de API. El complemento npd puede ejecutarse como un demonio o un DaemonSet.

Para obtener más información, consulte [node-problem-detector](#).

### Notas y restricciones

- Cuando utilice este complemento, no formatee ni particione los discos de nodo.
- Cada proceso de npd ocupa 30 mCPU y 100 MB de memoria.

### Descripción del permiso

Para monitorear los logs del núcleo, el complemento de npd necesita leer el host `/dev/kmsg`. Por lo tanto, el modo privilegiado debe estar habilitado. Para más detalles, consulte [Privilegiado](#).

Además, CCE mitiga los riesgos de acuerdo con el principio de privilegio mínimo. Solo están disponibles los siguientes privilegios para la ejecución de npd:

- `cap_dac_read_search`: permiso para acceder a `/run/log/journal`.
- `cap_sys_admin`: permiso para acceder a `/dev/kmsg`.

### Instalación del complemento

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster. Elija **Add-ons** en el panel de navegación, localice **npd** a la derecha y haga clic en **Install**.

**Paso 2** En la página **Install Add-on**, seleccione las especificaciones del complemento y establezca los parámetros relacionados.

- **Pods**: Establezca el número de pods en función de los requisitos de servicio.
- **Containers**: Seleccione una cuota de contenedor adecuada en función de los requisitos de servicio.

**Paso 3** Establezca los parámetros de npd y haga clic en **Install**.

Los parámetros son configurables solo en 1.16.0 y versiones posteriores. Para obtener más información, véase [Tabla 14-13](#).

---Fin

## Conceptos de comprobación de npd

### NOTA

Los conceptos de comprobación solo se admiten en 1.16.0 y versiones posteriores.

Compruebe los conceptos cubren eventos y estados.

- Relacionado con eventos

Para los elementos de comprobación relacionados con eventos, cuando se produce un problema, npd informa de un evento al servidor de API. El tipo de evento puede ser **Normal** (evento normal) o **Warning** (evento anormal).

**Tabla 14-7** Conceptos de comprobación relacionados con eventos

Concepto de comprobación	Función	Descripción
OOMKilling	Escuche los logs del núcleo y compruebe si ocurren eventos de OOM y si se informan.  Escenario típico: cuando el uso de memoria de un proceso en un contenedor excede el límite, OOM se activa y el proceso se termina.	Evento de advertencia Objeto de escucha: <b>/dev/kmsg</b>  Regla de coincidencia: "Killed process \\d+ (.+) total-vm:\\d+kB, anon-rss:\\d+kB, file-rss:\\d+kB.*"
TaskHung	Escuche los logs del núcleo y verifique si se producen y se informan los eventos de taskHung.  Escenario típico: La suspensión de E/S de disco provoca la suspensión del proceso.	Evento de advertencia Objeto de escucha: <b>/dev/kmsg</b>  Regla de coincidencia: "task \\S+:\\w+ blocked for more than \\w+ seconds\\."

Concepto de comprobación	Función	Descripción
ReadonlyFilesystem	<p>Compruebe si el error <b>Remount root filesystem read-only</b> ocurre en el kernel del sistema escuchando los logs del kernel.</p> <p>Escenario típico: Un usuario separa un disco de datos de un nodo por error en el ECS y las aplicaciones escriben datos continuamente en el punto de montaje del disco de datos. Como resultado, se produce un error de E/S en el núcleo y el disco se vuelve a montar como un disco de solo lectura.</p>	<p>Evento de advertencia</p> <p>Objeto de escucha: <b>/dev/kmsg</b></p> <p>Regla de coincidencia: <b>Remounting filesystem read-only</b></p>

- Relacionado con el estado

Para los elementos de comprobación relacionados con el estado, cuando se produce un problema, npd informa de un evento al servidor API y cambia el estado del nodo de forma síncrona. Esta función se puede utilizar junto con el [aislamiento de fallas del node-problem-controller](#) para aislar nodos.

**Si el período de comprobación no se especifica en los siguientes elementos de comprobación, el período predeterminado es de 30 segundos.**

**Tabla 14-8** Comprobación de componentes del sistema

Concepto de comprobación	Función	Descripción
Error de componente de red de contenedores CNIProblem	Comprobar el estado de los componentes CNI (componentes de red de contenedores).	No hay
Error de componente de tiempo de ejecución del contenedor CRIProblem	Compruebe el estado de Docker y containerd de los componentes CRI (componentes de tiempo de ejecución de contenedor).	Objeto de comprobación: Docker o containerd

Concepto de comprobación	Función	Descripción
Reinicios frecuentes de Kubelet FrequentKubeletRestart	Periódicamente haga retroceder los logs del sistema para comprobar si el componente clave de Kubelet se reinicia con frecuencia.	<ul style="list-style-type: none"> <li>● Umbral predeterminado: 10 reinicios en 10 minutos</li> <li>Si Kubelet se reinicia 10 veces en 10 minutos, indica que el sistema se reinicia con frecuencia y se genera una alarma de falla.</li> <li>● Objeto de escucha: logs en el directorio <b>/run/log/journal</b></li> </ul> <p><b>NOTA</b> Los sistemas operativos Ubuntu y HCE 2.0 no soportan los elementos de comprobación anteriores debido a formatos de log incompatibles.</p>
Reinicios frecuentes de Docker FrequentDockerRestart	Periódicamente haga retroceder los logs del sistema para comprobar si Docker en tiempo de ejecución de contenedor se reinicia con frecuencia.	
Reinicios frecuentes de containerd FrequentContainerdRestart	Periódicamente retroceder los logs del sistema para comprobar si containerd de tiempo de ejecución de contenedor se reinicia con frecuencia.	
kubelet error KubeletProblem	Compruebe el estado del componente clave de Kubelet.	No hay
Error de kube-proxy KubeProxyProblem	Compruebe el estado del componente clave de kube-proxy.	No hay

**Tabla 14-9** Comprobación de métricas del sistema

Concepto de comprobación	Función	Descripción
Tabla de conntrack llena ConntrackFullProblem	Compruebe si la tabla de conntrack está llena.	<ul style="list-style-type: none"> <li>● Umbral predeterminado: 90%</li> <li>● Uso: <b>nf_conntrack_count</b></li> <li>● Valor máximo: <b>nf_conntrack_max</b></li> </ul>
Recursos de disco insuficientes DiskProblem	Compruebe el uso del disco del sistema y los discos de datos de CCE (incluidos el disco lógico CRI y el disco lógico kubelet) en el nodo.	<ul style="list-style-type: none"> <li>● Umbral predeterminado: 90%</li> <li>● Fuente: df -h</li> </ul> <p>Actualmente, no se admiten discos de datos adicionales.</p>

Concepto de comprobación	Función	Descripción
Controladores de archivo insuficientes FDProblem	Compruebe si los controladores de archivo de FD están agotados.	<ul style="list-style-type: none"> <li>● Umbral predeterminado: 90%</li> <li>● Uso: el primer valor de <b>/proc/sys/fs/file-nr</b></li> <li>● Valor máximo: el tercer valor de <b>/proc/sys/fs/file-nr</b></li> </ul>
Memoria de nodo insuficiente MemoryProblem	Compruebe si la memoria está agotada.	<ul style="list-style-type: none"> <li>● Umbral predeterminado: 80%</li> <li>● Uso: <b>MemTotal-MemAvailable</b> en <b>/proc/meminfo</b></li> <li>● Valor máximo: <b>MemTotal</b> en <b>/proc/meminfo</b></li> </ul>
Recursos de proceso insuficientes PIDProblem	Compruebe si los recursos del proceso PID están agotados.	<ul style="list-style-type: none"> <li>● Umbral predeterminado: 90%</li> <li>● Uso: <b>nr_threads in /proc/loadavg</b></li> <li>● Valor máximo: valor menor entre <b>/proc/sys/kernel/pid_max</b> y <b>/proc/sys/kernel/threads-max</b>.</li> </ul>

**Tabla 14-10** Comprobación del almacenamiento

Concepto de comprobación	Función	Descripción
Disco de solo lectura DiskReadOnly	Realizar periódicamente pruebas de lectura y escritura en el disco del sistema y los discos de datos de CCE (incluidos el disco lógico de CRI y el disco lógico de Kubelet) del nodo para comprobar la disponibilidad de los discos clave.	<p>Rutas de detección:</p> <ul style="list-style-type: none"> <li>● <b>/mnt/paas/kubernetes/kubelet/</b></li> <li>● <b>/var/lib/docker/</b></li> <li>● <b>/var/lib/containerd/</b></li> <li>● <b>/var/paas/sys/log/cceaddon-npd/</b></li> </ul> <p>El archivo temporal <b>npd-disk-write-ping</b> se genera en la ruta de detección.</p> <p>Actualmente, no se admiten discos de datos adicionales.</p>

Concepto de comprobación	Función	Descripción
Recursos de disco insuficientes DiskProblem	Compruebe el uso del disco del sistema y los discos de datos de CCE (incluidos el disco lógico CRI y el disco lógico kubelet) en el nodo.	<ul style="list-style-type: none"> <li>● Umbral predeterminado: 90%</li> <li>● Fuente:  <code>df -h</code></li> </ul> Actualmente, no se admiten discos de datos adicionales.

Concepto de comprobación	Función	Descripción
<p>Error de grupo de almacenamiento de emptyDir</p> <p>EmptyDirVolumeGroupStatusError</p>	<p>Compruebe si el grupo de volúmenes efímeros en el nodo es normal.</p> <p>Impacto: el pod que depende del grupo de almacenamiento no puede escribir datos en el volumen temporal. El volumen temporal se vuelve a montar como un sistema de archivos de solo lectura por el kernel debido a un error de E/S.</p> <p>Escenario típico: al crear un nodo, un usuario configura dos discos de datos como un grupo de almacenamiento de volumen temporal. El usuario elimina algunos discos de datos por error. Como resultado, el grupo de almacenamiento se vuelve anormal.</p>	<ul style="list-style-type: none"> <li>● Periodo de detección: 30s</li> <li>● Fuente: <code>vgs -o vg_name, vg_attr</code></li> <li>● Principio: Compruebe si el VG (grupo de almacenamiento) está en el estado P. En caso afirmativo, se pierden algunos PV (discos de datos).</li> <li>● Programación conjunta: El planificador puede identificar automáticamente un error de grupo de almacenamiento de PV y evitar que los pods que dependen del grupo de almacenamiento se programen en el nodo.</li> </ul>
<p>Error del grupo de almacenamiento de PV</p> <p>LocalPvVolumeGroupStatusError</p>	<p>Compruebe el grupo de PV en el nodo.</p> <p>Impacto: los pods que dependen del grupo de almacenamiento no pueden escribir datos en el volumen persistente. El volumen persistente se vuelve a montar como un sistema de archivos de solo lectura por el kernel debido a un error de E/S.</p> <p>Escenario típico: Al crear un nodo, un usuario configura dos discos de datos como un grupo de almacenamiento de volúmenes persistentes. Algunos discos de datos se eliminan por error.</p>	<ul style="list-style-type: none"> <li>● Escenario excepcional: El complemento de npd no puede detectar la pérdida de todos los PV (discos de datos), lo que resulta en la pérdida de VG (grupos de almacenamiento). En este caso, kubelet aísla automáticamente el nodo, detecta la pérdida de VG (grupos de almacenamiento) y actualiza los recursos correspondientes de <b>nodestatus.allocatable</b> a <b>0</b>. Esto evita que los pods que dependen del grupo de almacenamiento se programen en el nodo. El daño de un PV solo no se puede detectar por este elemento de comprobación, sino por el elemento de comprobación <b>ReadOnlyFilesystem</b>.</li> </ul>

Concepto de comprobación	Función	Descripción
<p>Error de punto de montaje</p> <p>MountPointProblem</p>	<p>Compruebe el punto de montaje en el nodo.</p> <p>Definición excepcional: No se puede acceder al punto de montaje ejecutando el comando <b>cd</b>.</p> <p>Escenario típico: Network File System (NFS), por ejemplo, obsfs y s3fs se monta en un nodo. Cuando la conexión es anormal debido a excepciones del servidor de NFS de red o del mismo nivel, todos los procesos que acceden al punto de montaje se suspenden. Por ejemplo, durante una actualización del clúster, se reinicia un kubelet y se analizan todos los puntos de montaje. Si se detecta el punto de montaje anormal, la actualización falla.</p>	<p>Alternativamente, puede ejecutar el comando siguiente:</p> <pre>for dir in `df -h   grep -v "Mounted on"   awk '{print \\\$NF}'`;do cd \$dir; done &amp;&amp; echo "ok"</pre>
<p>E/S de disco suspendido</p> <p>DiskHung</p>	<p>Compruebe si la suspensión de E/S se produce en todos los discos del nodo, es decir, si no se responden las operaciones de lectura y escritura de E/S.</p> <p>Definición de suspensión de E/S: El sistema no responde a las solicitudes de E/S de disco y algunos procesos están en estado D.</p> <p>Escenario típico: Los discos no pueden responder debido a controladores de disco duro del sistema operativo anormales o a fallas graves en la red subyacente.</p>	<ul style="list-style-type: none"> <li>● Objeto de comprobación: todos los discos de datos</li> <li>● Fuente: /proc/diskstat</li> </ul> <p>Alternativamente, puede ejecutar el comando siguiente:</p> <pre>iostat -xmt 1</pre> <ul style="list-style-type: none"> <li>● Umbral:             <ul style="list-style-type: none"> <li>– Uso medio: ioutil &gt;= 0.99</li> <li>– Longitud media de la cola de E/S: avgqu-sz &gt;= 1</li> <li>– Volumen medio de transferencia de E/S: iops (w/s) + ioth (wMB/s) &lt;= 1</li> </ul> </li> </ul> <p><b>NOTA</b></p> <p>En algunos sistemas operativos, no hay cambios de datos durante la E/S. En este caso, calcule el uso de tiempo de E/S de CPU. El valor de iowait debe ser mayor que 0.8.</p>

Concepto de comprobación	Función	Descripción
E/S de disco lento DiskSlow	<p>Compruebe si todos los discos del nodo tienen E/S lentas, es decir, si las E/S responden lentamente.</p> <p>Escenario típico: los discos de EVS tienen E/S lentas debido a la fluctuación de la red.</p>	<ul style="list-style-type: none"> <li>● Objeto de comprobación: todos los discos de datos</li> <li>● Fuente: /proc/diskstat</li> </ul> <p>Alternativamente, puede ejecutar el comando siguiente:</p> <pre>iostat -xmt 1</pre> <ul style="list-style-type: none"> <li>● Umbral predeterminado: Latencia media de E/S: await &gt;= 5000 ms</li> </ul> <p><b>NOTA</b>                      Si las solicitudes de E/S no se responden y los datos <b>await</b> no se actualizan, este elemento de comprobación no es válido.</p>

**Tabla 14-11** Otros artículos de cheques

Concepto de comprobación	Función	Descripción
NTP anormal NTPProblem	<p>Compruebe si el servicio de sincronización de reloj de nodo ntpd o chronyd se está ejecutando correctamente y si se produce una desviación de tiempo del sistema.</p>	<p>Umbral de desplazamiento de reloj predeterminado: 8000 ms</p>
Error de proceso D ProcessD	<p>Compruebe si hay un proceso D en el nodo.</p>	<p>Umbral predeterminado: 10 procesos anormales detectados por tres veces consecutivas</p>
Error del proceso Z ProcessZ	<p>Compruebe si el nodo tiene procesos en estado Z.</p>	<p>Fuente:</p> <ul style="list-style-type: none"> <li>● /proc/{PID}/stat</li> <li>● Alternativamente, puede ejecutar el comando <b>ps aux</b>.</li> </ul> <p>Escenario excepcional: ProcessD ignora los procesos D residentes (latido del corazón y actualización) de los que depende el controlador de SDI en el nodo de BMS.</p>

Concepto de comprobación	Función	Descripción
Error de ResolvConf ResolvConfFileProblem	Compruebe si se ha perdido el archivo ResolvConf. Comprueba si el archivo ResolvConf es normal. Definición excepcional: No se incluye ningún servidor de resolución de nombres de dominio ascendente (servidor de nombres).	Objeto: <code>/etc/resolv.conf</code>
Evento programado existente ScheduledEvent	Compruebe si existen eventos de migración en vivo programados en el nodo. Un evento de plan de migración en vivo generalmente se desencadena por una falla de hardware y es un método automático de rectificación de fallas en la capa IaaS. Escenario típico: El host está defectuoso. Por ejemplo, el ventilador está dañado o el disco tiene sectores defectuosos. Como resultado, se activa la migración en vivo para las máquinas virtuales.	Fuente: <ul style="list-style-type: none"> <li>● <a href="http://169.254.169.254/meta-data/latest/events/scheduled">http://169.254.169.254/meta-data/latest/events/scheduled</a></li> </ul> Este elemento de verificación es una función Alfa y está deshabilitado de forma predeterminada.

El componente de kubelet tiene los siguientes elementos de comprobación predeterminados, que tienen errores o defectos. Puede solucionarlos actualizando el clúster o usando npd.

**Tabla 14-12** Conceptos de comprobación predeterminados de kubelet

Concepto de comprobación	Función	Descripción
Recursos de PID insuficientes PIDPressure	Compruebe si los PID son suficientes.	<ul style="list-style-type: none"> <li>● Intervalo: 10 segundos</li> <li>● Umbral: 90%</li> <li>● Defecto: En la versión de comunidad 1.23.1 y versiones anteriores, este concepto de comprobación no es válido cuando se utilizan más de 65535 PID. Para más detalles, véase el <a href="#">problema 107107</a>. En la versión de comunidad 1.24 y versiones anteriores, el hilo-max no se considera en este concepto de comprobación.</li> </ul>
Memoria insuficiente MemoryPressure	Compruebe si la memoria asignable para los contenedores es suficiente.	<ul style="list-style-type: none"> <li>● Intervalo: 10 segundos</li> <li>● Umbral: máx.100 MiB</li> <li>● Asignable = Memoria total de un nodo - Memoria reservada de un nodo</li> <li>● Defecto: Este concepto de comprobación comprueba solo la memoria consumida por contenedores y no la considera consumida por otros elementos del nodo.</li> </ul>
Recursos de disco insuficientes DiskPressure	Compruebe el uso del disco y el uso de inodes de los discos de kubelet y de Docker.	<ul style="list-style-type: none"> <li>● Intervalo: 10 segundos</li> <li>● Umbral: 90%</li> </ul>

## Aislamiento de fallas de node-problem-controller

### NOTA

El aislamiento de fallas solo es compatible con complementos de 1.16.0 y versiones posteriores.

De forma predeterminada, si varios nodos se vuelven defectuosos, NPC agrega manchas hasta un 10% de los nodos. Puede configurar `npc.maxTaintedNode` para aumentar el umbral.

El complemento de NPD de código abierto proporciona detección de fallas pero no aislamiento de fallas. CCE mejora el node-problem-controller (NPC) basado en el NPD de código abierto. Este componente se implementa en función del [controlador de nodo](#) de

Kubernetes. Para las fallas reportadas por NPD, NPC agrega automáticamente manchas a los nodos para el aislamiento de fallas de nodo.

**Tabla 14-13** Parámetros

Parámetro	Descripción	Predeterminado
npc.enable	Si habilitar NPC NPC no se puede deshabilitar en 1.18.0 o versiones posteriores.	true
npc.maxTaintedNode	Verifique cuántos nodos pueden agregar manchas npc para mitigar el impacto cuando se produce una única falla en varios nodos. El formato int y el formato porcentual son compatibles.	10% Rango de valores: <ul style="list-style-type: none"> <li>● El valor está en formato int y varía de 1 a infinito.</li> <li>● El valor varía de 1% a 100%, en porcentaje. El valor mínimo de este parámetro multiplicado por el número de nodos de clúster es 1.</li> </ul>
npc.affinity	Afinidad de nodos del responsable del tratamiento	N/A

## Consulta de eventos de NPD

Los eventos reportados por el complemento de npd se pueden consultar en la página **Nodes**.

**Paso 1** Inicie sesión en la consola de CCE.

**Paso 2** Haga clic en el nombre del clúster para acceder a la consola del clúster. Elija **Nodes** en el panel de navegación.

**Paso 3** Busque la fila que contiene el nodo de destino y haga clic en **View Events**.

**Figura 14-2** Consulta de eventos de nodo

Pods (Allocated/...	CPU Request/Li...	Memory Request/Li...	Runtime Version & OS Version	Billing Mode	Operation
12 / 60	34.13% 70.8%	36.69% 57.97%	docker://18.9.0 CentOS Linux 7 (Core)	Pay-per-use	Monitor <b>View Events</b> More ▾

Puede consultar eventos en la página mostrada.

**Events** ×

💡 Event data is stored only for one hour and then automatically cleared.

Start Date -- End Date Enter a Kubernetes event name

Kubernet...	Event ...	Occurr...	Event Name	Kubernetes Event	First Occurred	Last Occurred
yangtse-contro...	Alarm	211	Abnormal	Failed to add route: {cce 172.21.0.128/...	Aug 24, 2022 08:28:0...	Aug 27, 2022 13:58:0...
yangtse-contro...	Normal	934	Normal	Try to add route {cce 172.21.0.128/25 0...	Aug 24, 2022 08:14:2...	Aug 27, 2022 13:58:0...
yangtse-contro...	Alarm	302	Abnormal	Failed to add route: {cce 172.21.0.128/...	Aug 24, 2022 08:38:1...	Aug 27, 2022 13:48:0...
yangtse-contro...	Alarm	107	Abnormal	Failed to add route: {cce 172.21.0.128/...	Aug 24, 2022 09:58:0...	Aug 27, 2022 13:38:0...

----Fin

## Alarmas de AOM

Para los elementos de comprobación de NPD relacionados con el estado, puede configurar Application Operations Management (AOM) para convertir estados anormales en alarmas de AOM y notificarlo por mensaje SMS o correo electrónico.

**Paso 1** Inicie sesión en la consola de AOM.

**Paso 2** En el panel de navegación, elija **Alarm Center** > **Alarm Rules** y haga clic en **Create Alarm Rule**.

**Paso 3** Establezca una regla de alarma.

- **Rule Type:** Seleccione **Threshold alarm**.
- **Monitored Object:** Seleccione **Command input**.
- Escriba `sum(problem_gauge{clusterName="test"}) by (podIP,type)` en el cuadro de texto.

Alarm Rule Settings

Rule Type: Threshold alarm Event alarm

\* Monitored Object: Select resource objects Command input

× | Q | ? Statistical Period: 1 ...



No data available.

- **Alarm Condition:** Activar una alarma importante si el valor promedio es mayor o igual a 1 por una vez consecutiva en un período de monitorización.

\* Alarm Condition Custom

Trigger Condition In  if the    for  a Major alarm will be generated.

---

Advanced Settings ^

Alarm Clearance If the monitored object does not meet the trigger condition for  the alarm will be automatically cleared.

Action Taken for Insufficient Data

- (Opcional) **Alarm notification**: Para recibir notificaciones de alarmas por correo electrónico o mensaje SMS, configure las reglas de acción para la regla de alarma. Si no hay ninguna regla de acción disponible, puede crear una.

----Fin

## Recopilación de métricas de Prometheus

El pod daemon de NPD expone los datos métricos de Prometheus en el puerto 19901. De forma predeterminada, el pod de NPD se agrega con la anotación **metrics.alpha.kubernetes.io/custom-endpoints:'[{"api":"prometheus","path":"/metrics","port":"19901","names":""}]'**. Puede crear un colector de Prometheus para identificar y obtener métricas de NPD desde **http://{{NpdPodIP}}:{{NpdPodPort}}/metrics**.

### NOTA

Si la versión del complemento npd es anterior a 1.16.5, el puerto expuesto de las métricas de Prometheus es **20257**.

Actualmente, los datos métricos incluyen **problem\_counter** y **problem\_gauge** como se muestra a continuación.

```
# HELP problem_counter Number of times a specific type of problem have occurred.
# TYPE problem_counter counter
problem_counter{reason="DockerHung"} 0
problem_counter{reason="DockerStart"} 0
problem_counter{reason="EmptyDirVolumeGroupStatusError"} 0
...
# HELP problem_gauge Whether a specific type of problem is affecting the node or not.
# TYPE problem_gauge gauge
problem_gauge{reason="CNIIIsDown",type="CNIPProblem"} 0
problem_gauge{reason="CNIIIsUp",type="CNIPProblem"} 0
problem_gauge{reason="CRIIsDown",type="CRIPProblem"} 0
problem_gauge{reason="CRIIsUp",type="CRIPProblem"} 0
..
```

## Historial de cambios

**Tabla 14-14** Versiones de complementos de CCE

Versión del complemento	Versión de clúster admitida	Característica actualizada	Versión de la comunidad (solo para clústeres de v1.17 y posteriores)
1.17.4	/v1.(17 19 21 23 25).*/	● Optimizado el elemento de comprobación de DiskHung.	<b>0.8.10</b>

Versión del complemento	Versión de clúster admitida	Característica actualizada	Versión de la comunidad (solo para clústeres de v1.17 y posteriores)
1.17.3	/v1.(17 19 21 23 25).*/	<ul style="list-style-type: none"> <li>● Admitida la configuración por porcentaje para el número máximo de nodos taint que se pueden agregar al NPC</li> <li>● Agregado ProcessZ.</li> <li>● Optimizado el elemento de comprobación de NTPProblem para detectar la desviación de tiempo.</li> <li>● Corregidos los procesos residentes en el estado D (existen en el nodo BMS).</li> </ul>	<b>0.8.10</b>
1.17.2	/v1.(17 19 21 23 25).*/	<ul style="list-style-type: none"> <li>● Agregado DiskHung.</li> <li>● Agregado DiskSlow.</li> <li>● Agregado ProcessD.</li> <li>● Agregado MountPointProblem.</li> <li>● Para evitar conflictos con el rango de puertos de servicio, el puerto de escucha de comprobación de estado predeterminado se cambia a <b>19900</b> y el puerto de exposición métrica de Prometheus predeterminado a <b>19901</b>.</li> <li>● Clústeres compatibles de v1.25.</li> </ul>	<b>0.8.10</b>

Versión del complemento	Versión de clúster admitida	Característica actualizada	Versión de la comunidad (solo para clústeres de v1.17 y posteriores)
1.16.4	/v1.(17 19 21 23).*/	<ul style="list-style-type: none"> <li>● Agregado el elemento de comprobación beta ScheduledEven y se usó en la API de metaData para detectar eventos de migración de VM en frío y en vivo causados por excepciones de máquina host. Este elemento de verificación está deshabilitado de forma predeterminada.</li> </ul>	<b>0.8.10</b>
1.16.3	/v1.(17 19 21 23).*/	<ul style="list-style-type: none"> <li>● Agregada la función de comprobar el archivo de configuración ResolvConf.</li> </ul>	<b>0.8.10</b>
1.16.1	/v1.(17 19 21 23).*/	<ul style="list-style-type: none"> <li>● Agregado node-problem-controller. Aislamiento básico de fallas soportado.</li> <li>● Agregados los elementos de comprobación PID, FD, disco, memoria, agrupación temporal de volúmenes y agrupación PV.</li> </ul>	<b>0.8.10</b>
1.15.0	/v1.(17 19 21 23).*/	<ul style="list-style-type: none"> <li>● Endurecimiento integral de elementos de control para evitar falsos positivos</li> <li>● Comprobación del kernel compatible. Informes apoyados de eventos OOMKilled y TaskHung.</li> </ul>	<b>0.8.10</b>
1.14.11	/v1.(17 19 21).*/	<ul style="list-style-type: none"> <li>● Clústeres CCE compatibles de v1.21.</li> </ul>	<b>0.7.1</b>
1.14.5	/v1.(17 19).*/	<ul style="list-style-type: none"> <li>● Se solucionó el problema de que no se podían obtener métricas de supervisión.</li> </ul>	<b>0.7.1</b>
1.14.4	/v1.(17 19).*/		<b>0.7.1</b>

Versión del complemento	Versión de clúster admitida	Característica actualizada	Versión de la comunidad (solo para clústeres de v1.17 y posteriores)
1.14.2	/v1.(17 19).*/	<ul style="list-style-type: none"> <li>● Se proporcionó soporte para Kubernetes 1.19, sistema operativo Ubuntu y contenedores Kata.</li> </ul>	<b>0.7.1</b>
1.13.8	/v1.15.11 v1.17.*/	<ul style="list-style-type: none"> <li>● Se ha corregido el problema de comprobación de estado del CNI en la red del túnel contenedor.</li> <li>● Cuotas de recursos ajustadas.</li> </ul>	<b>0.7.1</b>
1.13.6	/v1.15.11 v1.17.*/	<ul style="list-style-type: none"> <li>● Se ha corregido el problema de que los procesos zombis no se recuperaban.</li> </ul>	<b>0.7.1</b>
1.13.5	/v1.15.11 v1.17.*/	<ul style="list-style-type: none"> <li>● Agregada la configuración de tolerancia a la corrosión.</li> </ul>	<b>0.7.1</b>
1.13.2	/v1.15.11 v1.17.*/	<ul style="list-style-type: none"> <li>● Agregadas las restricciones de recursos y se mejoró la capacidad de detección del complemento CNI.</li> </ul>	<b>0.7.1</b>

## 14.5 dashboard

### Presentación

Kubernetes Dashboard es una interfaz de usuario de propósito general basada en web para clústeres de Kubernetes. Permite a los usuarios gestionar aplicaciones que se ejecutan en el clúster y solucionar problemas, así como gestionar el clúster en sí, mediante la ejecución de comandos.

Con Kubernetes Dashboard, puede:

- Desplegar las aplicaciones en contenedores en un clúster de Kubernetes.
- Diagnosticar problemas de aplicaciones en contenedores.
- Gestionar los recursos del clúster.
- Ver las aplicaciones que se ejecutan en un clúster.
- Crear y modificar los recursos de Kubernetes (como Deployments, trabajos y DaemonSets).

- Comprobar los errores que se producen en un clúster.

Por ejemplo, puede escalar una Deployments, realizar una actualización continua, reiniciar un pod o desplegar una nueva aplicación.

Comunidad de código abierto: <https://github.com/kubernetes/dashboard>

## Instalación del complemento

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster. Elija **Add-ons** en el panel de navegación, localice **dashboard** a la derecha y haga clic en **Install**.

**Paso 2** En la página **Configuration**, configure los siguientes parámetros:

- **Certificate Configuration**: Configure un certificado para el panel de control.
  - Uso de una certificación personalizada
    - **Certificate File**: Haga clic en  para ver el archivo de certificado de ejemplo.
    - **Private Key**: Haga clic en  para ver la clave privada de ejemplo.
  - Uso de un certificado predeterminado

### AVISO

El certificado predeterminado generado por el panel de control no es válido, lo que afecta al acceso normal al panel de control con un navegador. Se recomienda cargar manualmente un certificado válido para que el navegador pueda verificar su acceso y proteger su conexión.

**Paso 3** Haga clic en **Install**.

----Fin

## Acceso al complemento del panel de control

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster. Elija **Add-ons** en el panel de navegación. En la página que se muestra, compruebe que el complemento del panel de control está en estado **Running** y haga clic en **Access**.

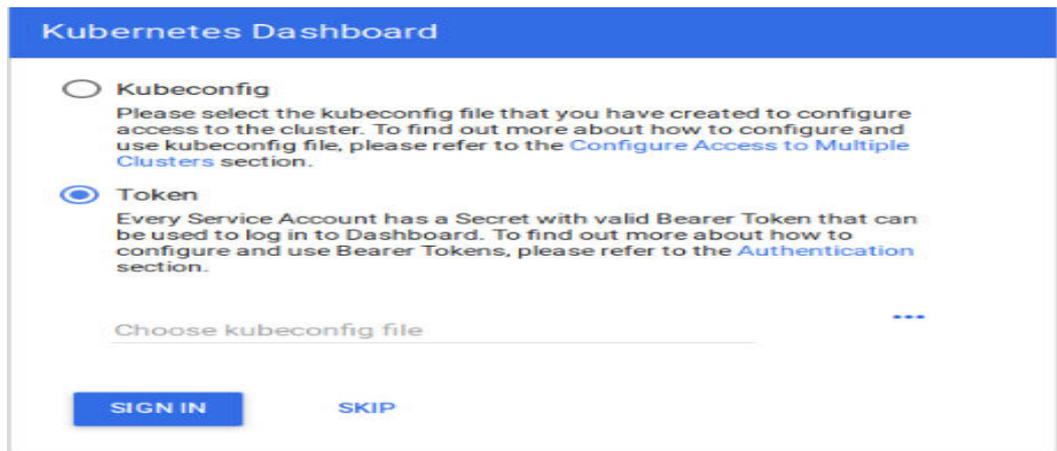
**Paso 2** Copie el token en el cuadro de diálogo que se muestra.

**Paso 3** En la página de inicio de sesión del panel de control, seleccione **Token**, pegue el token copiado y haga clic en **SIGN IN**.

### NOTA

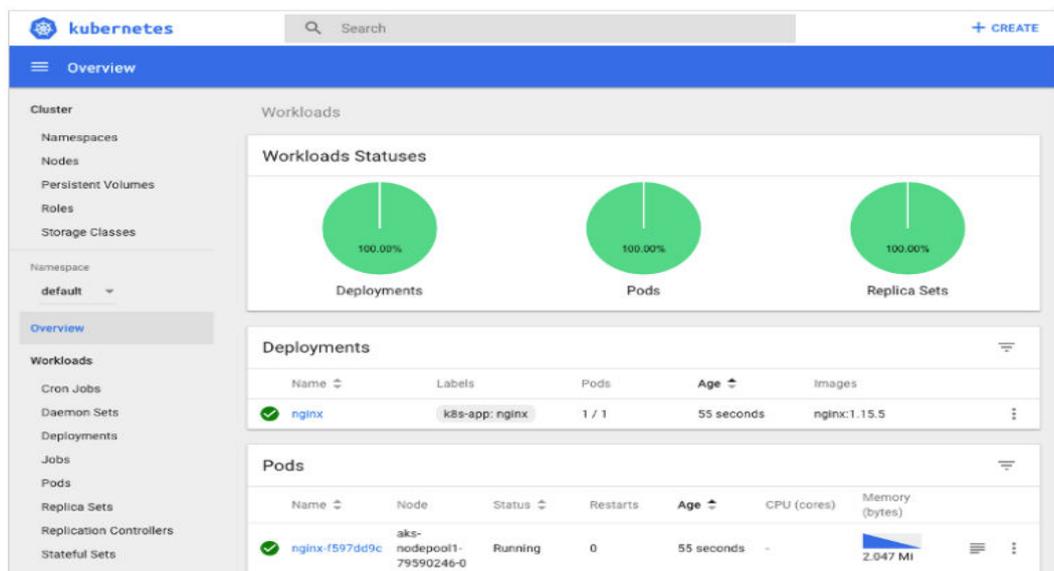
De forma predeterminada, este complemento no admite el inicio de sesión usando kubeconfig autenticado por certificado. Se recomienda utilizar el modo de token para iniciar sesión. Para obtener más información, véase <https://github.com/kubernetes/dashboard/issues/2474#issuecomment-348912376>.

Figura 14-3 Inicio de sesión de token



Paso 4 Vea la página del panel de control como se muestra en Figura 14-4.

Figura 14-4 Página del panel de control



----Fin

## Modificación de permisos

Después de instalar el panel, el rol inicial solo puede ver la mayoría de los recursos que se muestran en el panel. Para solicitar los permisos para realizar otras operaciones en el panel, debe modificar los recursos de autorización de RBAC en segundo plano.

### Procedimiento

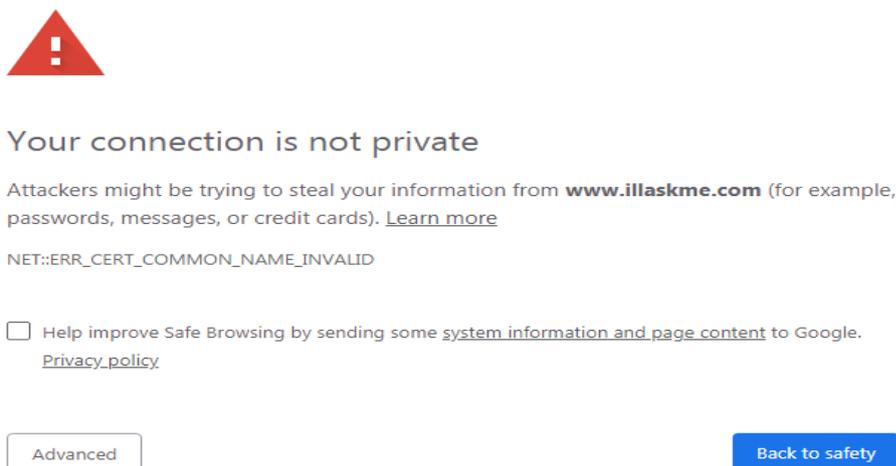
Modifique la regla **kubernetes-dashboard-minimal** en el ClusterRole.

Para obtener más información sobre cómo usar la autorización RBAC, visite <https://kubernetes.io/docs/reference/access-authn-authz/rbac/>.

## Solución de problemas de acceso

Cuando se utiliza Google Chrome para acceder al panel de control, se muestra el mensaje de error "ERR\_CERT\_INVALID", en lugar de la página de inicio de sesión. La posible causa es que el certificado predeterminado generado por el panel de control no pasa la verificación de Google Chrome. Hay dos soluciones a este problema:

**Figura 14-5** Mensaje de error mostrado en Google Chrome



- Solución 1: Utilice el navegador Firefox para acceder al panel de control. En el área **Exceptions** de la página **Proxy Settings**, agregue la dirección del panel a las direcciones que omitirán el servidor proxy. A continuación, se mostrará la página de inicio de sesión del panel de control.
- Solución 2: Inicie Google Chrome con el indicador **--ignore-certificate-errors** para ignorar el error del certificado.

Windows: Guarde la dirección del panel de control. Cierre todas las ventanas activas de Google Chrome. Presione la tecla Windows + R para mostrar el cuadro de diálogo **Run**. Escriba **chrome --ignore-certificate-errors** en el cuadro de diálogo **Run** para abrir una nueva ventana de Google Chrome. En la barra de direcciones, introduzca la dirección del panel de control para abrir la página de inicio de sesión.

## Historial de cambios

**Tabla 14-15** Versiones de complementos de CCE

Versión del complemento	Versión de clúster admitida	Versión de la comunidad (solo para clústeres de v1.17 y posteriores)
2.2.3	/v1.(21 23 25).*/	<b>2.7.0</b>
2.1.1	/v1.(19 21 23).*/	<b>2.5.0</b>
2.0.10	/v1.(15 17 19 21).*/	<b>2.0.0</b>

Versión del complemento	Versión de clúster admitida	Versión de la comunidad (solo para clústeres de v1.17 y posteriores)
2.0.4	/v1.(15 17 19).*/	<b>2.0.0</b>
2.0.3	/v1.(15 17 19).*/	<b>2.0.0</b>
2.0.2	/v1.(17 19).*/	<b>2.0.0</b>
2.0.1	/v1.(15 17).*/	<b>2.0.0</b>
2.0.0	/v1.(17).*/	<b>2.0.0</b>

## 14.6 autoscaler

### Presentación

Autoscaler es un importante controlador de Kubernetes. Es compatible con el ajuste de microservicios y es clave para el diseño sin servidor.

Cuando el uso de CPU o memoria de un microservicio es demasiado alto, se activa el escalado automático de pods horizontal para agregar pods para reducir la carga. Estos pods se pueden reducir automáticamente cuando la carga es baja, lo que permite que el microservicio funcione de la manera más eficiente posible.

CCE simplifica la creación, actualización y ajuste manual de clústeres de Kubernetes, en los que las cargas de tráfico cambian con el tiempo. Para equilibrar el uso de recursos y el rendimiento de las cargas de trabajo de los nodos, Kubernetes introduce el complemento de escalado automático para cambiar el tamaño de un clúster automáticamente en función del uso de recursos requerido para las cargas de trabajo implementadas en el clúster. Para obtener más información, véase [Creación de una política del ajuste de nodos](#).

Comunidad de código abierto <https://github.com/kubernetes/autoscaler>

### Cómo funciona el complemento

autoscaler controla la expansión y la reducción automáticas.

- **Expansión automática**

Puede elegir cualquiera de los siguientes métodos:

- Si los pods de un clúster no se pueden programar debido a nodos de trabajo insuficientes, se activa el ajuste del clúster para agregar nodos. Los nodos que se van a agregar tienen la misma especificación que la configurada para el grupo de nodos al que pertenecen los nodos.

La expansión automática se realizará cuando:

- Los recursos del nodo son insuficientes.
- No se establece ninguna política de afinidad de nodo en la configuración de programación de pod. Es decir, si un nodo se ha configurado como un nodo de afinidad para pods, no se agregará automáticamente ningún nodo cuando los

pods no se puedan programar. Para obtener más información acerca de cómo configurar la política de afinidad de nodo, consulte [Política de programación \(afinidad/antiafinidad\)](#).

- Cuando el clúster cumple con la política de ajuste de nodo, también se activa la expansión del clúster. Para obtener más información, véase [Creación de una política del ajuste de nodos](#).

#### NOTA

El complemento sigue la política "No Menos, No Más". Por ejemplo, si se requieren tres núcleos para crear un pod y el sistema admite nodos de cuatro núcleos y ocho núcleos, el escalador automático creará preferentemente un nodo de cuatro núcleos.

#### ● **Reducción automática**

Cuando un nodo de clúster está inactivo durante un período de tiempo (10 minutos de forma predeterminada), se activa la ampliación del clúster y el nodo se elimina automáticamente. Sin embargo, no se puede eliminar un nodo de un clúster si existen los siguientes pods:

- Pods que no cumplen con los requisitos específicos establecidos en Pod Disruption Budgets ([PodDisruptionBudget](#))
- Pods que no se pueden programar para otros nodos debido a restricciones como las políticas de afinidad y antiafinidad
- Pods que tienen la anotación **cluster-autoscaler.kubernetes.io/safe-to-evict: 'false'**
- Pods (excepto aquellos creados por DaemonSets en el espacio de nombres del sistema kube) que existen en el espacio de nombres del sistema kube en el nodo
- Pods que no son creados por el controlador (Deployment/ReplicaSet/job/StatefulSet)

#### NOTA

Cuando un nodo cumple con las condiciones de la reducción, el autoscaler agrega la mancha **DeletionCandidateOfClusterAutoscaler** al nodo por adelantado para evitar que los pods se programen en el nodo. Después de desinstalar el complemento de autoscaler, si la mancha todavía existe en el nodo, elimínelo manualmente.

## Notas y restricciones

- Solo los clústeres de v1.9.7-r1 y posteriores admiten el escalador automático.
- Asegúrese de que hay suficientes recursos para instalar el complemento.
- El autoscaler solo puede agregar o quitar **los nodos de VM de pago por uso**.
- El grupo de nodos predeterminado no admite el ajuste automático. Para obtener más información, véase [Descripción del DefaultPool](#).

## Instalación del complemento

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster. Elija **Add-ons** en el panel de navegación, localice **autoscaler** a la derecha y haga clic en **Install**.

**Paso 2** Configure los parámetros de instalación del complemento.

**Tabla 14-16** Configuración de las especificaciones

Parámetro	Descripción
Add-on Specifications	<p>El complemento se puede desplegar en las siguientes especificaciones:</p> <p><b>NOTA</b></p> <p>Cuando el complemento del autoscaler se despliega en modo HA o personalizado, existen las políticas de antiafinidad entre las instancias de complemento y las instancias de complemento se despliegan en diferentes nodos. Por lo tanto, el número de nodos disponibles en el clúster debe ser mayor o igual que el número de instancias de complemento para garantizar una alta disponibilidad del complemento.</p> <ul style="list-style-type: none"> <li>● <b>Single:</b> El complemento se despliega con un solo pod.</li> <li>● <b>HA50:</b> El complemento se despliega con dos pods, sirviendo a un clúster con 50 nodos y asegurando una alta disponibilidad.</li> <li>● <b>HA200:</b> El complemento se despliega con dos pods, sirviendo a un clúster con 50 nodos y asegurando una alta disponibilidad. Cada pod utiliza más recursos que los de la especificación <b>HA50</b>.</li> <li>● <b>Custom:</b> Puede personalizar el número de pods y las especificaciones según sea necesario.</li> </ul>

**Tabla 14-17** Configuración de parámetros

Parámetro	Descripción
Scaling	<p>Puede seleccionar las siguientes opciones según sea necesario:</p> <ul style="list-style-type: none"> <li>● <b>Los nodos se agregan automáticamente (desde el grupo de nodos) cuando no se pueden programar los pods del clúster.</b>                      Es decir, cuando un pod está en estado <b>Pending</b>, se realiza una expansión automática. Si un nodo se ha configurado como un nodo de afinidad para los pods, no se agregará automáticamente ningún nodo cuando los pods no se puedan programar. En general, una política de HPA funciona con tal ajuste. Para obtener más información, véase <a href="#">Uso de HPA y CA para el ajuste automático de cargas de trabajo y nodos</a>.</li> </ul> <p>Si este parámetro no está seleccionado, el ajuste solo se puede realizar con <a href="#">las políticas de ajuste de nodos</a>.</p> <ul style="list-style-type: none"> <li>● Reducción automática de nodo                             <ul style="list-style-type: none"> <li>– <b>Node Idle Time (min):</b> Tiempo durante el cual un nodo no debe ser necesario antes de que sea elegible para escalar hacia abajo. Valor predeterminado: 10 minutos.</li> <li>– <b>Scale-in Threshold:</b> Si el porcentaje de CPU y memoria solicitada en un nodo está por debajo de este umbral, se activará la reducción automática de escala para eliminar el nodo del clúster. El valor predeterminado es 0.5, lo que significa 50%.</li> <li>– <b>Stabilization Window (s)</b>                                      ¿Cuánto tiempo después de una expansión se reanuda una evaluación de reducción. Valor predeterminado: 10 minutos.</li> </ul> </li> </ul> <p><b>NOTA</b></p> <p>Si existen ambas expansión y reducción automáticas en un clúster, se recomienda establecer <b>How long after a expansión that a reducción evaluation resumes</b> en 0 minutos. Esto puede evitar que la reducción horizontal de nodo se bloquee debido a la expansión continua de algunos grupos de nodos o reintentos tras un fallo de expansión que resulta en un desperdicio inesperado de recursos de nodo.</p> <p>Cuánto tiempo después de la eliminación del nodo se reanuda una evaluación de reducción. Valor predeterminado: 10 minutos.</p> <p>Cuánto tiempo después de un fallo de reducción se reanuda una evaluación de reducción. Valor predeterminado: 3 minutos. Para obtener detalles sobre el impacto y la relación entre los intervalos de refrigeración reducción configurados en el grupo de nodos y el autoscaler, consulte <a href="#">Descripción del periodo de enfriamiento a escala</a>.</p>

Parámetro	Descripción
	<ul style="list-style-type: none"> <li>– <b>Max. Nodes for Batch Deletion:</b> Número máximo de nodos vacíos que se pueden eliminar al mismo tiempo. Valor predeterminado: 10. Esta característica solo se aplica a los nodos inactivos. Los nodos inactivos se pueden escalar simultáneamente. Los nodos que no están inactivos solo se pueden escalar uno por uno.</li> </ul> <p><b>NOTA</b>                      Durante la reducción del nodo, si el pod del nodo no necesita ser desalojado (como los pods de DaemonSet), el nodo está inactivo. De lo contrario, el nodo no está inactivo.</p> <ul style="list-style-type: none"> <li>– <b>Check Interval:</b> Intervalo para comprobar de nuevo un nodo que no se podía quitar antes. Valor predeterminado: 5 minutos.</li> </ul>
Total Nodes	Número máximo de nodos que puede gestionar el clúster, dentro del cual se realiza la expansión horizontal del clúster.
Total CPUs	Suma máxima de núcleos de CPU de todos los nodos de un clúster, dentro del cual se realiza la expansión horizontal del clúster.
Total Memory (GB)	Suma máxima de memoria de todos los nodos de un clúster, dentro del cual se realiza la expansión horizontal del clúster.

**Paso 3** Cuando se complete la configuración, haga clic en **Install**.

---Fin

## Descripción del periodo de enfriamiento a escala

Los intervalos de enfriamiento de escalado se pueden configurar en la configuración del grupo de nodos y en la configuración del complemento del autoscaler.

### Intervalo de enfriamiento de reducción configurado en un grupo de nodos

Este intervalo indica el período durante el cual no se pueden eliminar los nodos agregados al grupo de nodos actual después de una operación de expansión. Este intervalo tiene efecto en el nivel del grupo de nodos.

### Intervalo de enfriamiento de reducción configurado en el complemento del autoscaler

El intervalo después de una expansión indica el período durante el cual no se puede escalar todo el clúster después de que el complemento de autoscaler active expansión (debido a los pods, las métricas y las políticas de ajuste no programables). Este intervalo tiene efecto a nivel de clúster.

El intervalo después de eliminar un nodo indica el período durante el cual no se puede escalar el clúster después de que el complemento de autoscaler active reducción. Este intervalo tiene efecto a nivel de clúster.

El intervalo después de una reducción fallida indica el período durante el cual el clúster no se puede escalar después de que el complemento de autoscaler active la reducción. Este intervalo tiene efecto a nivel de clúster.

## Historial de cambios

**Tabla 14-18** Versiones de complementos de CCE

<b>Versión del complemento</b>	<b>Versión de clúster admitida</b>	<b>Versión de la comunidad (solo para clústeres de v1.17 y posteriores)</b>
1.25.21	/v1.25.*/ 	<b>1.25.0</b>
1.25.11	/v1.25.*/ 	<b>1.25.0</b>
1.25.7	/v1.25.*/ 	<b>1.25.0</b>
1.23.31	/v1.23.*/ 	<b>1.23.0</b>
1.23.21	/v1.23.*/ 	<b>1.23.0</b>
1.23.17	/v1.23.*/ 	<b>1.23.0</b>
1.23.10	/v1.23.*/ 	<b>1.23.0</b>
1.23.9	/v1.23.*/ 	<b>1.23.0</b>
1.23.8	/v1.23.*/ 	<b>1.23.0</b>
1.23.7	/v1.23.*/ 	<b>1.23.0</b>
1.23.3	/v1.23.*/ 	<b>1.23.0</b>
1.21.29	/v1.21.*/ 	<b>1.21.0</b>
1.21.20	/v1.21.*/ 	<b>1.21.0</b>
1.21.16	/v1.21.*/ 	<b>1.21.0</b>
1.21.9	/v1.21.*/ 	<b>1.21.0</b>
1.21.8	/v1.21.*/ 	<b>1.21.0</b>
1.21.6	/v1.21.*/ 	<b>1.21.0</b>
1.21.4	/v1.21.*/ 	<b>1.21.0</b>
1.21.2	/v1.21.*/ 	<b>1.21.0</b>
1.21.1	/v1.21.*/ 	<b>1.21.0</b>
1.19.35	/v1.19.*/ 	<b>1.19.0</b>
1.19.27	/v1.19.*/ 	<b>1.19.0</b>
1.19.22	/v1.19.*/ 	<b>1.19.0</b>
1.19.14	/v1.19.*/ 	<b>1.19.0</b>
1.19.13	/v1.19.*/ 	<b>1.19.0</b>
1.19.12	/v1.19.*/ 	<b>1.19.0</b>

Versión del complemento	Versión de clúster admitida	Versión de la comunidad (solo para clústeres de v1.17 y posteriores)
1.19.11	/v1.19.*	<b>1.19.0</b>
1.19.9	/v1.19.*	<b>1.19.0</b>
1.19.8	/v1.19.*	<b>1.19.0</b>
1.19.7	/v1.19.*	<b>1.19.0</b>
1.19.6	/v1.19.*	<b>1.19.0</b>
1.19.3	/v1.19.*	<b>1.19.0</b>
1.17.27	/v1.17.*	<b>1.17.0</b>
1.17.22	/v1.17.*	<b>1.17.0</b>
1.17.21	/v1.17.*	<b>1.17.0</b>
1.17.19	/v1.17.*	<b>1.17.0</b>
1.17.17	/v1.17.*	<b>1.17.0</b>
1.17.16	/v1.17.*	<b>1.17.0</b>
1.17.15	/v1.17.*	<b>1.17.0</b>
1.17.14	/v1.17.*	<b>1.17.0</b>
1.17.8	/v1.17.*	<b>1.17.0</b>
1.17.7	/v1.17.*	<b>1.17.0</b>
1.17.5	/v1.17.*	<b>1.17.0</b>
1.17.2	/v1.17.*	<b>1.17.0</b>

## 14.7 nginx-ingress

### Presentación

Kubernetes utiliza kube-proxy para exponer Services y proporcionar balanceo de carga. La implementación es en la capa de transporte. Cuando se trata de aplicaciones de Internet, donde se genera un bucket de información, el reenvío debe estar más detallado, controlado de manera precisa y flexible por políticas y balanceadores de carga para ofrecer un mayor rendimiento.

Aquí es donde entran las entradas. Los ingresos proporcionan funciones de reenvío de la capa de aplicación, como hosts virtuales, balanceo de carga, proxy SSL y enrutamiento HTTP, para Services a los que se puede acceder directamente fuera de un clúster.

Kubernetes ha lanzado oficialmente el controlador de ingreso basado en Nginx. `nginx-ingress` es un complemento que utiliza ConfigMaps para almacenar configuraciones de Nginx. El controlador de entrada de Nginx genera configuraciones de Nginx para una entrada y escribe las configuraciones en el pod de Nginx con la API de Kubernetes. Estas configuraciones se pueden modificar y actualizar mediante recarga.

El complemento de `nginx-ingress` en CCE se implementa usando el gráfico y la imagen de comunidad de código abierto. CCE no mantiene el complemento. Por lo tanto, no se recomienda que el complemento de `nginx-ingress` se use comercialmente.

Puede visitar la [comunidad de código abierto](#) para obtener más información.

#### NOTA

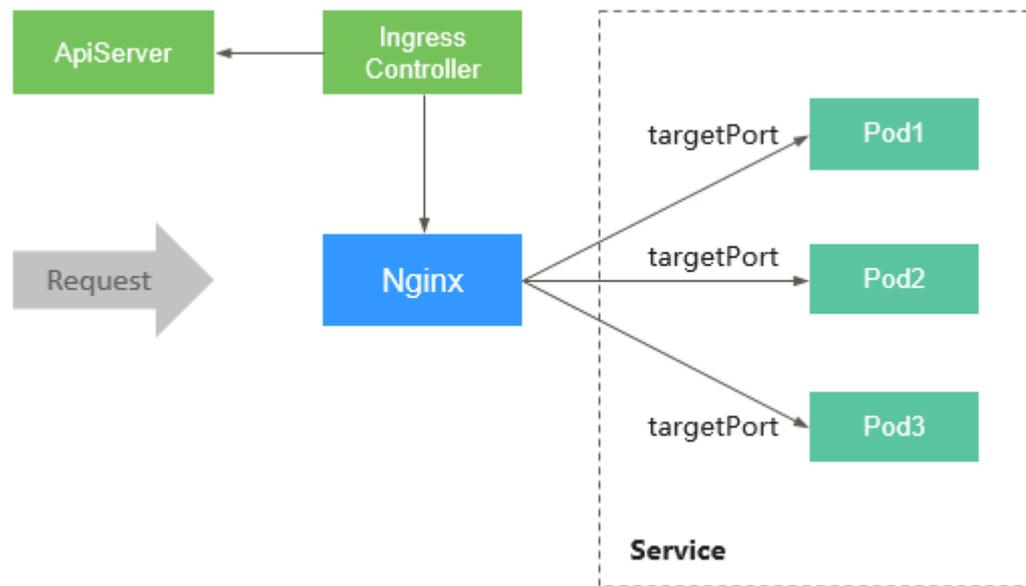
- Al instalar el complemento, puede agregar configuraciones definiendo la configuración de Nginx. Las configuraciones tienen efecto globalmente. Este parámetro se genera configurando el archivo `nginx.conf` y afecta a todas las entradas gestionadas. Puede buscar los parámetros relacionados en el [ConfigMap](#). Si los parámetros configurados no se incluyen en las opciones enumeradas en ConfigMap, las configuraciones no surten efecto.
- Después de instalar el complemento, puede conectarse con Nginx y agregar **annotations** de Kubernetes a una entrada específica para personalizar su comportamiento al crear una entrada en la consola de CCE. Para obtener más información sobre el campo de **annotations** de Kubernetes, consulte [Anotaciones](#).
- No modifique o elimine manualmente el balanceador de carga y el oyente que son creados automáticamente por CCE. De lo contrario, la carga de trabajo será anormal. Si los ha modificado o eliminado por error, debe desinstalar el complemento de `nginx-ingress` y volver a instalarlo.

## Cómo funciona `nginx-ingress`

`nginx-ingress` consiste en el objeto de ingreso, el controlador de ingreso y Nginx. El controlador de ingreso ensambla las entradas en el archivo de configuración de Nginx (`nginx.conf`) y recarga Nginx para hacer que las configuraciones cambiadas surtan efecto. Cuando detecta que el pod en un Service cambia, cambia dinámicamente la configuración del grupo de servidores ascendentes de Nginx. En este caso, el proceso de Nginx no necesita ser recargado. [Figura 14-6](#) muestra cómo funciona `nginx-ingress`.

- Una entrada es un grupo de reglas de acceso que reenvía solicitudes a los servicios especificados en función de nombres de dominio o direcciones URL. Las entradas se almacenan en el servicio de almacenamiento de objetos etcd, y se agregan, eliminan, modifican y consultan con las API.
- El controlador de entrada supervisa los cambios de objetos de recursos como entradas, Services, puntos de conexión, secretos (principalmente certificados y claves TLS), nodos y ConfigMaps en tiempo real y realiza automáticamente operaciones en Nginx.
- Nginx implementa el balanceo de carga y control de acceso en la capa de aplicación.

Figura 14-6 Principios de funcionamiento de nginx-ingress



## Restricciones

- Este complemento solo se puede instalar en clústeres de CCE de v1.15 o posterior.
- **kubernetes.io/ingress.class: "nginx"** debe agregarse a la anotación de la entrada creada invocando a una API.
- Los balanceadores de carga dedicados deben ser del tipo de red (TCP/UDP) que admita las redes privadas (con una IP privada).
- El nodo donde nginx-ingress-controller se está ejecutando y los contenedores que se está ejecutando en el nodo no puede acceder a Nginx Ingress. En este caso, realice el despliegue de antiafinidad para las cargas de trabajo y nginx-ingress-controlador. Para obtener más información, véase [Despliegue de antiafinidad para cargas de trabajo y nginx-ingress-controller](#).

## Requisitos previos

Antes de crear una carga de trabajo, debe tener un clúster disponible. Si no hay ningún clúster disponible, cree uno según [Compra de un clúster de CCE](#).

## Instalación del complemento

### 📖 NOTA

- La vulnerabilidad de [CVE-2021-25746](#) se ha corregido en nginx-ingress-controller de v1.2.0 (correspondiente al complemento 2.1.0 de CCE nginx-ingress). [Las reglas](#) se agregan para deshabilitar algunas anotaciones propensas al acceso no autorizado.
- La vulnerabilidad de [CVE-2021-25745](#) se ha corregido en nginx-ingress-controller de v1.2.0 (correspondiente al complemento 2.1.0 de CCE nginx-ingress). [Las reglas](#) se agregan para deshabilitar algunas rutas de acceso propensas al acceso no autorizado.

**Paso 1** Inicie sesión en la consola de CCE.

**Paso 2** En el panel de navegación de la izquierda, elija **Add-ons** y haga clic en **Install** en nginx-ingress.

**Paso 3** Establezca los parámetros de configuración.

- **Cluster:** Seleccione un clúster.
- **Specifications:** Seleccione o personalice las especificaciones adicionales según sea necesario.
  - **Pods:** Establezca el número de pods en función de los requisitos de servicio.
  - **Containers:** Establezca una cuota de contenedor adecuada en función de los requisitos de servicio.
- **Load Balancer:** Seleccione un balanceador de carga compartido o dedicado. Si no hay ningún balanceador de carga disponible, cree uno primero. El balanceador de carga tiene al menos dos oyentes, y los puertos 80 y 443 no están ocupados por oyentes.
- **Nginx Parameters:** La configuración del archivo **nginx.conf** afectará a todas las entradas gestionadas. Puede buscar parámetros relacionados con **ConfigMaps**. Si los parámetros que ha configurado no están incluidos en las opciones enumeradas en esos ConfigMaps, los parámetros no tendrán efecto.

Por ejemplo, puede utilizar el parámetro **keep-alive-requests** para describir cómo establecer el número máximo de solicitudes para mantener las conexiones activas en 100.

```
{
  "keep-alive-requests": "100"
}
```

- **Default 404 Service:** De forma predeterminada, se utiliza el servicio 404 proporcionado por el complemento. Para personalizar el servicio 404, escriba el nombre del espacio de nombres/servicio. Si el servicio no existe, se producirá un error en la instalación del complemento.

**Paso 4** Haga clic en **Install**.

----Fin

## Despliegue de antiafinidad para cargas de trabajo y nginx-ingress-controller

El nodo donde nginx-ingress-controller se está ejecutando y los contenedores que se está ejecutando en el nodo no puede acceder a Nginx Ingress. Para evitar este problema, debe configurar una regla antiafinidad para indicar al planificador que no coubique la carga de trabajo y nginx-ingress-controller en el mismo nodo.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx:alpine
        imagePullPolicy: IfNotPresent
        name: nginx
        imagePullSecrets:
```

```

- name: default-secret
affinity:
  podAntiAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
      - labelSelector:
          matchExpressions: # Use the labels of nginx-ingress-controller
            to implement anti-affinity.
              - key: app
                operator: In
                values:
                  - nginx-ingress
              - key: component
                operator: In
                values:
                  - controller
    namespaces:
      - kube-system
    topologyKey: kubernetes.io/hostname
    
```

## Historial de cambios

Tabla 14-19 Versiones de complementos de CCE

Versión del complemento	Versión de clúster admitida	Versión de la comunidad (solo para clústeres de v1.17 y posteriores)
2.2.3	/v1.25.*/*	<b>1.5.1</b>
2.2.1	/v1.25.*/*	<b>1.5.1</b>
2.1.3	/v1.(19 21 23).*/	<b>1.2.1</b>
2.1.1	/v1.(19 21 23).*/	<b>1.2.1</b>
2.1.0	/v1.(19 21 23).*/	<b>1.2.0</b>
2.0.1	/v1.(19 21 23).*/	<b>1.1.1</b>
1.3.2	/v1.(15 17 19 21).*/	<b>0.49.3</b>
1.2.6	/v1.(15 17 19).*/	<b>0.46.0</b>
1.2.5	/v1.(15 17 19).*/	<b>0.46.0</b>
1.2.3	/v1.(15 17 19).*/	<b>0.43.0</b>
1.2.2	/v1.(15 17).*/	<b>0.43.0</b>

## 14.8 metrics-server

A partir de la versión 1.8, Kubernetes proporciona métricas de uso de recursos, como el uso contenedor de CPU y memoria, con la API de métricas. Los usuarios pueden acceder directamente a estas métricas (por ejemplo, mediante el comando **kubectl top**) o las pueden utilizar los controladores (por ejemplo, Horizontal Pod Autoscaler) en un clúster para la toma de decisiones. El componente específico es el servidor de métricas, que se utiliza para sustituir a heapster para proporcionar las funciones similares. heapster ha sido abandonado gradualmente desde la v1.11.

metrics-server es un agregador para monitorear los datos de los recursos del clúster central. Puede instalar rápidamente este complemento en la consola de CCE.

Después de instalar metrics-server, puede crear una política de HPA en la página de ficha **Workload Scaling** de la página **Auto Scaling**. Para obtener más información, véase [Creación de una política de HPA para el escalado automático de cargas de trabajo](#).

El proyecto y la documentación oficial de la comunidad están disponibles en <https://github.com/kubernetes-sigs/metrics-server>.

## Instalación del complemento

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster. Elija **Add-ons** en el panel de navegación, localice **metrics-server** a la derecha y haga clic en **Install**.

**Paso 2** Seleccione **Single**, **Custom** o **HA** para **Add-on Specifications**.

- **Pods:** Establezca el número de pods en función de los requisitos de servicio.
- **Containers:** Establezca una cuota de contenedor adecuada en función de los requisitos de servicio.

**Paso 3** Haga clic en **Install**.

---Fin

## Historial de cambios

**Tabla 14-20** Versiones de complementos de CCE

Versión del complemento	Versión de clúster admitida	Community Version (for Only Clusters of v1.17 and Later)
1.3.2	/v1.(19 21 23 25).*/	<b>0.4.4</b>
1.2.1	/v1.(19 21 23).*/	<b>0.4.4</b>
1.1.10	/v1.(15 17 19 21).*/	<b>0.4.4</b>
1.1.4	/v1.(15 17 19).*/	<b>0.4.4</b>
1.1.2	/v1.(15 17 19).*/	<b>0.4.4</b>
1.1.1	/v1.(13 15 17 19).*/	<b>0.3.7</b>
1.1.0	/v1.(13 15 17 19).*/	<b>0.3.7</b>
1.0.5	/v1.13.* v1.15.* v1.17.*/	<b>0.3.7</b>

## 14.9 cce-hpa-controller

cce-hpa-controller es un complemento desarrollado por CCE, que se puede utilizar para escalar de manera flexible las Deployments basadas en métricas como el uso de la CPU y el uso de la memoria.

Después de instalar este complemento, puede crear una política de CustomedHPA en la página de ficha **Workload Scaling** de la página **Auto Scaling**. Para obtener más información, véase [Creación de una política de CustomedHPA para el ajuste automático de cargas de trabajo](#).

## Características principales

- El ajuste se puede realizar basándose en el porcentaje del número actual de pods.
- Se puede establecer la etapa del ajuste mínimo.
- Se pueden realizar diferentes operaciones de ajuste según los valores métricos reales.

## Notas y restricciones

- Este complemento solo se puede instalar en clústeres de CCE de v1.15 o posterior.
- Si la versión cce-hpa-controller es anterior a 1.2.11, el complemento [prometheus](#) debe estar instalado. Si la versión de cce-hpa-controller es 1.2.11 o posterior, los complementos que pueden proporcionar API de métricas deben estar instalados. Seleccione uno de los siguientes complementos según la versión del clúster y los requisitos reales.
  - [metrics-server](#): proporciona las métricas básicas de uso de recursos, como CPU de contenedor y uso de memoria. Es compatible con todas las versiones de clúster.
  - [prometheus](#): proporciona las métricas personalizadas además de las métricas básicas de recursos. Es necesario registrar Prometheus como el servicio que proporciona API de métricas. Para obtener más información, véase [Proporcionar métricas de recursos](#). Este complemento solo admite clústeres de v1.21 o anteriores.
  - [kube-prometheus-stack](#): proporciona las métricas personalizadas además de las métricas básicas de recursos. Es necesario registrar Prometheus como el servicio que proporciona API de métricas. Para obtener más información, véase [Proporcionar métricas de recursos](#). Este complemento solo admite clústeres de v1.23 o posterior.

## Instalación del complemento

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster. Elija **Add-ons** en el panel de navegación, localice **cce-hpa-controller** a la derecha y haga clic en **Install**.

**Paso 2** Seleccione **Single** o **Custom** para **Add-on Specifications**.

### NOTA

Los complementos de instancia única se utilizan solo para la verificación del servicio. En los despliegues comerciales, seleccione **Custom** según las especificaciones del clúster. Las especificaciones de cce-hpa-controller se deciden por el número total de contenedores en el clúster y el número de políticas de ajuste. Se recomienda configurar 500m CPU y 1,000 MiB de memoria por cada 5,000 contenedores y 100m CPU y 500 MiB de memoria por cada 1,000 políticas de ajuste.

- **Pods**: Establezca el número de pods en función de los requisitos de servicio.
- **Containers**: Establezca una cuota de contenedor adecuada en función de los requisitos de servicio.

**Paso 3** Haga clic en **Install**.

----Fin

## Historial de cambios

**Tabla 14-21** Versiones de complementos de CCE

Versión del complemento	Versión de clúster admitida
1.3.3	/v1.(19 21 23 25).*/
1.3.1	/v1.(19 21 23).*/
1.2.12	/v1.(15 17 19 21).*/
1.2.11	/v1.(15 17 19 21).*/
1.2.10	/v1.(15 17 19 21).*/
1.2.4	/v1.(15 17 19).*/
1.2.3	/v1.(15 17 19).*/
1.2.2	/v1.(15 17 19).*/
1.2.1	/v1.(15 17 19).*/
1.1.3	/v1.(15 17).*/

## 14.10 prometheus

### Presentación

Prometheus es un marco de monitoreo y alerta de sistema de código abierto. Se deriva del sistema de monitoreo borgmon de Google, que fue creado por ex empleados de Google que trabajaban en SoundCloud en 2012. Prometheus fue desarrollado como un proyecto comunitario de código abierto y lanzado oficialmente en 2015. En 2016, Prometheus se unió oficialmente a la Cloud Native Computing Foundation, después de Kubernetes.

CCE le permite instalar rápidamente Prometheus como complemento.

Página web oficial de Prometheus: <https://prometheus.io/>

Comunidad de código abierto: <https://github.com/prometheus/prometheus>

### Notas y restricciones

El complemento de prometheus solo se admite en clústeres de v1.21 y anteriores.

### Funciones

Como marco de supervisión de próxima generación, Prometheus tiene las siguientes características:

- Potente modelo de datos multidimensional
  - a. Los datos de series temporales se identifican por el nombre de la métrica y el par clave-valor.

- b. Se pueden establecer etiquetas multidimensionales para todas las métricas.
  - c. Los modelos de datos no requieren cadenas de caracteres separadas por puntos.
  - d. Los modelos de datos se pueden agregar, cortar y cortar.
  - e. Se admite el formato de punto flotante doble. Todas las etiquetas se pueden establecer en unicode.
- Instrucción de consulta flexible y potente (PromQL): Una instrucción de consulta admite la adición, la multiplicación y la conexión para múltiples métricas.
  - Fácil de gestionar: El servidor de Prometheus es un archivo binario independiente que puede funcionar localmente. No depende del almacenamiento distribuido.
  - Eficiente: Cada punto de muestreo ocupa solo 3.5 bytes, y un servidor de Prometheus puede procesar millones de métricas.
  - El modo de extracción se utiliza para recopilar datos de series de tiempo, lo que facilita las pruebas locales y evita que los servidores defectuosos empujen métricas incorrectas.
  - Los datos de series temporales se pueden enviar al servidor de Prometheus en el modo de gateway push.
  - Los usuarios pueden obtener los destinos supervisados mediante la detección de servicios o la configuración estática.
  - Múltiples GUI visuales están disponibles.
  - Fácil de escalar

Como los datos recopilados pueden perderse, Prometheus no es aplicable si existe un alto requisito de exactitud de los datos recopilados. Sin embargo, Prometheus tiene grandes ventajas de consulta si se utiliza para registrar datos de series temporales. Además, Prometheus es aplicable a la arquitectura de microservicios.

## Instalación del complemento

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster. Elija **Add-ons** en el panel de navegación, localice **prometheus** a la derecha y haga clic en **Install**.

**Paso 2** En el paso **Configuration**, establezca los siguientes parámetros:

**Tabla 14-22** parámetros del complemento prometheus

Parámetro	Descripción
Add-on Specifications	<p>Seleccione una especificación adicional basada en los requisitos de servicio. Las opciones son las siguientes:</p> <ul style="list-style-type: none"> <li>● <b>Demo(&lt;= 100 contenedores)</b>: El tipo de especificación es aplicable al entorno de demostración de experiencia y función. En esta memoria descriptiva, Prometheus ocupa pocos recursos pero tiene capacidades de procesamiento limitadas. Se recomienda utilizar esta especificación cuando el número de contenedores en el clúster no exceda de 100.</li> <li>● <b>Small(&lt;= 2000 contenedores)</b>: se recomienda utilizar esta especificación cuando el número de contenedores en el clúster no exceda 2,000.</li> <li>● <b>Medium(&lt;= 5000 contenedores)</b>: se recomienda utilizar esta especificación cuando el número de contenedores en el clúster no exceda 5,000.</li> <li>● <b>Large(&gt; 5000 contenedores)</b>: se recomienda utilizar esta especificación cuando el número de contenedores en el clúster supera a 5,000.</li> </ul>
Instances	Número de pods que se crearán para que coincidan con las especificaciones del complemento seleccionado. El número no se puede modificar.
Container	Cuotas de CPU y memoria del contenedor permitidas para las especificaciones de complemento seleccionadas. Las cuotas no se pueden modificar.
Data Retention (days)	Número de días para almacenar datos de monitoreo personalizados. El valor predeterminado es 15 días.
Storage	<p>Los discos duros en la nube se pueden utilizar como almacenamiento. Establezca los siguientes parámetros según se le solicite:</p> <ul style="list-style-type: none"> <li>● <b>AZ</b>: Establezca este parámetro en función de los requisitos del sitio. Una AZ es una región física donde los recursos utilizan las fuentes de alimentación y las redes independientes. Las AZ están físicamente aisladas, pero se interconectan a través de una red interna.</li> <li>● <b>Disk Type</b>: Se admiten E/S comunes, E/S altas y E/S ultraaltas.</li> <li>● <b>Capacity</b>: Introduzca la capacidad de almacenamiento en función de los requisitos de servicio. El valor predeterminado es 10 GB.</li> </ul> <p><b>NOTA</b>                      Si ya existe un PVC en la supervisión del espacio de nombres, el almacenamiento configurado se utilizará como origen de almacenamiento.</p>

**Paso 3** Haga clic en **Install**. Después de la instalación, el complemento despliega las siguientes instancias en el clúster.

- prometheus-operator: despliega y gestiona el Prometheus Server basado en CRD (CustomResourceDefinitions), y monitoriza y procesa los eventos relacionados con estos CRD. Es el centro de control de todo el sistema.
- prometheus (servidor): un clúster de servidor de Prometheus desplegado por el operador basado en los CRD de Prometheus que puede considerarse StatefulSets.
- prometheus-kube-state-metrics: convierte los datos de la métrica de Prometheus en un formato que puede ser identificado por las API de Kubernetes.
- custom-metrics-apiserver: agrega métricas personalizadas al servidor nativo de la API de Kubernetes.
- prometheus-node-exporter: desplegado en cada nodo para recopilar datos de monitorización de nodos.
- grafana: visualiza los datos de monitorización.

----Fin

## Proporcionar métricas de recursos

Las métricas de recursos de contenedores y nodos, como el uso de CPU y memoria, se pueden obtener con la API de métricas de Kubernetes. Se puede acceder directamente a las métricas de recursos, por ejemplo, mediante el comando **kubectl top**, o utilizar las políticas HPA o CustomedHPA personalizadas para el ajuste automático.

El complemento puede proporcionar la API de Kubernetes Metrics que está deshabilitada de forma predeterminada. Para habilitar la API, cree el siguiente objeto de APIService:

```
apiVersion: apiregistration.k8s.io/v1
kind: APIService
metadata:
  labels:
    app: custom-metrics-apiserver
    release: cceaddon-prometheus
    name: v1beta1.metrics.k8s.io
spec:
  group: metrics.k8s.io
  groupPriorityMinimum: 100
  insecureSkipTLSVerify: true
  service:
    name: custom-metrics-apiserver
    namespace: monitoring
    port: 443
  version: v1beta1
  versionPriority: 100
```

Puede guardar el objeto como un archivo, nombrarlo **metrics-apiservice.yaml** y ejecutar el siguiente comando:

```
kubectl create -f metrics-apiservice.yaml
```

Ejecute el comando **kubectl top**. Si se muestra la siguiente información, se puede acceder a la API de métricas:

```
# kubectl top pod -n monitoring
NAME                                                    CPU (cores)
MEMORY (bytes)
.....
custom-metrics-apiserver-d4f556ff9-12j2m             38m          44Mi
.....
```

**AVISO**

Para desinstalar el complemento, ejecute el siguiente comando kubectl y elimine el objeto APIService. De lo contrario, el complemento de metrics-server no se puede instalar debido a los recursos APIService residuales.

```
kubectl delete APIService v1beta1.metrics.k8s.io
```

**Referencia**

- Para obtener más información sobre los conceptos y configuraciones de Prometheus, consulte la [Documentación oficial de Prometheus](#).
- Para obtener más información acerca de cómo instalar Node Exporter, consulte el [node\\_exporter GitHub](#).
- Para obtener más información sobre cómo enviar mensajes de Slack, consulta [Webhooks entrantes](#).

**Historial de cambios**

**Tabla 14-23** Versiones de complementos de CCE

Versión del complemento	Versión de clúster admitida	Versión de la comunidad (solo para clústeres de v1.17 y posteriores)
2.23.32	/v1.(17 19 21).*/	<b>2.10.0</b>
2.23.31	/v1.15.*/	<b>2.10.0</b>
2.23.30	/v1.(17 19 21).*/	<b>2.10.0</b>
2.21.14	/v1.(17 19 21).*/	<b>2.10.0</b>
2.21.12	/v1.15.*/	<b>2.10.0</b>
2.21.11	/v1.(17 19).*/	<b>2.10.0</b>
1.15.1	/v1.(15 17).*/	<b>2.10.0</b>

**14.11 web-terminal**

El complemento web-terminal es un servidor terminal ligero que le permite usar kubectl en la interfaz de usuario web. Proporciona una interfaz remota de línea de comandos (CLI) a través de navegador web y HTTP, y se puede integrar fácilmente en un sistema independiente. Puede acceder directamente al complemento como servicio para obtener información e iniciar sesión en un servidor con cmdb.

web-terminal puede ejecutarse en todos los sistemas operativos soportados por Node.js y no depende de los módulos locales. Es rápido y fácil de instalar y soporta múltiples sesiones.

Comunidad de código abierto: <https://github.com/rabchev/web-terminal>

## Notas y restricciones

- Este complemento solo se puede instalar en los clústeres de v1.21 o anteriores. Los clústeres de Arm no son compatibles.
- Este complemento se encuentra actualmente en fase beta. Cloudshell reemplazará este complemento en el futuro.
- Al instalar web-terminal para usar kubectl, debe iniciar sesión con su cuenta en la nube o como usuario de IAM con el permiso de CCE Administrator. Para obtener más información acerca de cómo controlar el permiso de kubectl, consulte [Control de permisos de web-terminal](#).
- El complemento web-terminal solo se puede usar después de que coredns se instale en un clúster.

## Precauciones

El complemento web-terminal se puede utilizar para gestionar los clústeres de CCE. Mantenga la contraseña de inicio de sesión segura para evitar un funcionamiento inesperado.

## Instalación del complemento

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster. Elija **Add-ons** en el panel de navegación, localice **web-terminal** a la derecha y haga clic en **Install**.

**Paso 2** Configure los parámetros siguientes:

- **Access Mode:** El valor se fija a **NodePort**. El complemento de web-terminal se accede en modo NodePort de forma predeterminada y solo se puede usar si cualquier nodo del clúster tiene una EIP. Si se selecciona este tipo de acceso, una EIP debe estar vinculada al clúster donde se instalará web-terminal.
- **Username:** El valor predeterminado es **root** y no se puede cambiar.
- **Password:** contraseña para iniciar sesión en web-terminal. Mantenga segura la contraseña. El complemento web-terminal se puede utilizar para gestionar los clústeres de CCE. Mantenga la contraseña de inicio de sesión segura para evitar un funcionamiento inesperado.
- **Confirm Password:** Ingrese la contraseña de nuevo.

**Paso 3** Haga clic en **Install**.

----Fin

## Conexión a un clúster mediante el complemento de web-terminal

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster. Elija **Add-ons** en el panel de navegación.

**Paso 2** Encuentre **web-terminal** a la derecha y haga clic en **Access**.

----Fin

## Control de permisos de web-terminal

Después de instalar web-terminal, kubectl utiliza el cluster-admin de ClusterRole de forma predeterminada y puede operar los recursos de Kubernetes en el clúster. Si necesita cambiar

manualmente a otro ClusterRole puede ejecutar **kubectl edit clusterrolebinding web-terminal** para modificar el ServiceAccount de web-terminal.

Para obtener más información acerca de ClusterRole y ClusterRoleBinding en [Permisos de espacio de nombres \(basados en Kubernetes RBAC\)](#).

**AVISO**

- Los permisos de terminal web configurados manualmente se podrían restablecer después de actualizar el complemento de terminal web. Se recomienda hacer una copia de respaldo de las configuraciones antes de la actualización.
- Antes de usar kubectl para modificar ClusterRoleBindings asegúrese de que kubectl ha sido configurado con los permisos requeridos.

## Historial de cambios

Tabla 14-24 Versiones de complementos de CCE

Versión del complemento	Versión de clúster admitida	Versión de la comunidad (solo para clústeres de v1.17 y posteriores)
1.1.12	/v1.(15 17 19 21).*/	<b>0.6.6</b>
1.1.6	/v1.(15 17 19).*/	<b>0.6.6</b>
1.1.5	/v1.(15 17 19).*/	<b>0.6.6</b>
1.1.3	/v1.(17 19).*/	<b>0.6.6</b>
1.0.6	/v1.(15 17).*/	<b>0.6.6</b>
1.0.5	/v1.(9 11 13 15 17).*/	<b>0.6.6</b>

## 14.12 gpu-device-plugin (anteriormente gpu-beta)

### Presentación

gpu-device-plugin (anteriormente gpu-beta) es un complemento de gestión de dispositivos que admite GPU de contenedores. Si se utilizan nodos de GPU en el clúster, este complemento debe estar instalado.

### Notas y restricciones

- El controlador que se va a descargar debe ser un archivo **.run**.
- Solo se admiten los controladores de NVIDIA Tesla, no los controladores de GRID.
- Al instalar o reinstalar el complemento, asegúrese de que la dirección de descarga del controlador sea correcta y accesible. CCE no verifica la validez de la dirección.

- El complemento `gpu-device-plugin` solo permite descargar el controlador y ejecutar el script de instalación. El estado del complemento solo indica cómo se está ejecutando el complemento, no si el controlador se ha instalado correctamente.
- Si utiliza nodos de multi-GPU A100 o de A800, debe instalar manualmente el servicio `nvidia-fabricmanager` que coincida con la versión del controlador. Para obtener más información, véase [Instalación del servicio de nvidia-fabricmanager](#).

## Instalación del complemento

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster. Elija **Add-ons** en el panel de navegación, localice **gpu-device-plugin** a la derecha y haga clic en **Install**.

**Paso 2** Configure el enlace del controlador.

### AVISO

- Si el enlace de descarga es una dirección de red pública, por ejemplo `https://us.download.nvidia.com/tesla/470.103.01/NVIDIA-Linux-x86_64-470.103.01.run` una EIP a cada nodo de GPU. Para obtener más información sobre cómo obtener el enlace del controlador, consulte [Obtención del enlace del conductor de la red pública](#).
- Si el enlace de descarga es un URL de OBS, no es necesario vincular una EIP a los nodos de la GPU. Para obtener más información sobre cómo obtener el enlace del controlador, consulte [Obtención del enlace del conductor de OBS](#).
- Asegúrese de que la versión del controlador de NVIDIA coincida con el nodo de la GPU.
- Después de cambiar la versión del controlador, reinicie el nodo para que el cambio surta efecto.
- Para los sistemas del kernel de Linux 5.x, como Huawei Cloud EulerOS 2.0 o Ubuntu 22.04, se recomienda utilizar el controlador 470 o posterior.

**Paso 3** Haga clic en **Install**.

----**Fin**

## Comprobación del complemento

Después de instalar el complemento, ejecute el comando `nvidia-smi` en el nodo de GPU y el contenedor que programa los recursos de GPU para verificar la disponibilidad del dispositivo y el controlador de GPU.

Nodo de GPU:

```
cd /opt/cloud/cce/nvidia/bin && ./nvidia-smi
```

Contenedor:

```
cd /usr/local/nvidia/bin && ./nvidia-smi
```

Si se devuelve la información de la GPU, el dispositivo estará disponible y el complemento se instalará correctamente.

```

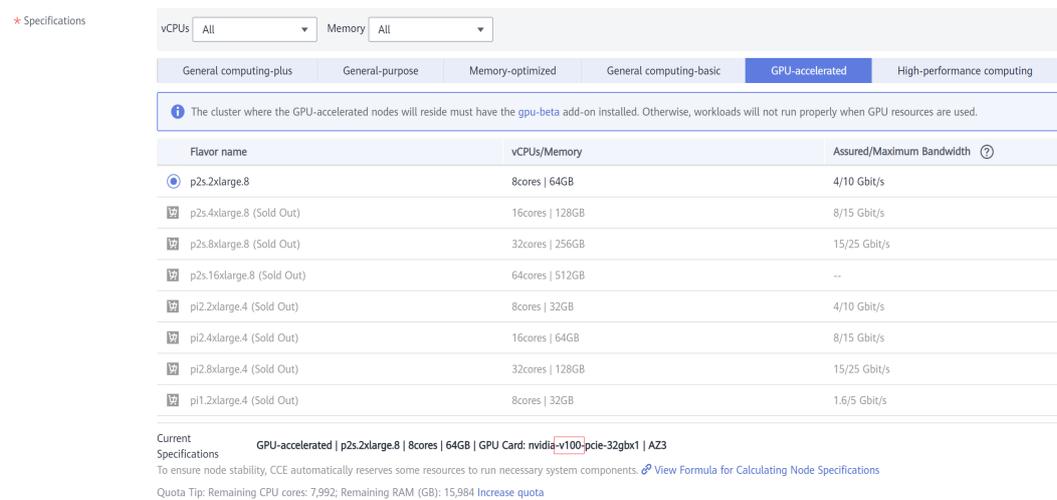
+-----+
| NVIDIA-SMI 440.118.02    Driver Version: 440.118.02    CUDA Version: 10.2    |
+-----+-----+
| GPU Name      Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
+-----+-----+-----+
|   0   31C   P0   23W / 300W |  0MiB / 16160MiB |      0%    Default  |
+-----+-----+-----+

+-----+
| Processes:
| GPU      PID  Type  Process name                               GPU Memory
|-----|-----|-----|-----|-----|
| No running processes found
+-----+
    
```

## Obtención del enlace del conductor de la red pública

- Paso 1** Inicie sesión en la consola de CCE.
- Paso 2** Haga clic en **Create Node** y seleccione el nodo de GPU que se creará en el área **Specifications**. El modelo de tarjeta de GPU del nodo se muestra en la parte inferior de la página.

**Figura 14-7** Consulta del modelo de tarjeta de GPU



- Paso 3** Visite <https://www.nvidia.com/Download/Find.aspx?lang=en>.
- Paso 4** Seleccione la información del controlador en la página **NVIDIA Driver Downloads** como se muestra en **Figura 14-8**. **Operating System** debe ser **Linux 64-bit**.

Figura 14-8 Configuración de parámetros

**NVIDIA Driver Downloads**

Official Advanced Driver Search | NVIDIA

**Product Type:** Data Center / Tesla

**Product Series:** V-Series

**Product:** Tesla V100

**Operating System:** Linux 64-bit

**CUDA Toolkit:** Any

**Language:** English (US)

**Recommended/Beta:** All

Search

Click the Search button to perform your search.

**Paso 5** Después de confirmar la información del controlador, haga clic en **SEARCH**. Se muestra una página en la que se muestra la información del conductor, como se muestra en **Figura 14-9**. Haga clic en **DOWNLOAD**.

Figura 14-9 Información del conductor

**Data Center Driver For Linux X64**

**Version:** 470.103.01

**Release Date:** 2022.1.31

**Operating System:** Linux 64-bit

**CUDA Toolkit:** 11.4

**Language:** English (US)

**File Size:** 259.86 MB

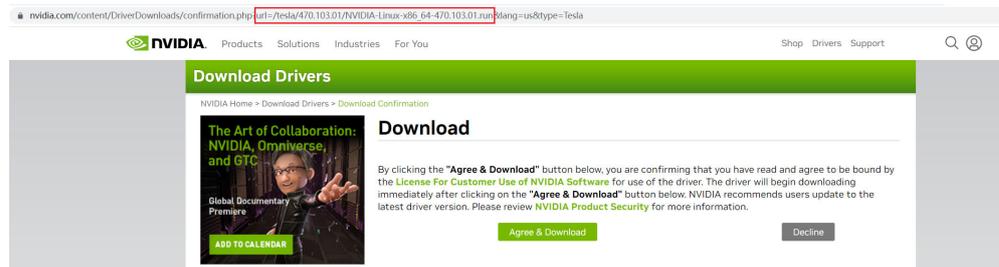
Download

Release Highlights	Supported Products	Additional Information
Release notes, supported GPUs and other documentation can be found at: <a href="https://docs.nvidia.com/datacenter/tesla/index.html">https://docs.nvidia.com/datacenter/tesla/index.html</a>		

**Paso 6** Obtenga el enlace del controlador de cualquiera de las siguientes maneras:

- Método 1: Como se muestra en **Figura 14-10**, busque `url=/tesla/470.103.01/NVIDIA-Linux-x86_64-470.103.01.run` en el cuadro de dirección del navegador. A continuación, complementar para obtener el enlace conductor [https://us.download.nvidia.com/tesla/470.103.01/NVIDIA-Linux-x86\\_64-470.103.01.run](https://us.download.nvidia.com/tesla/470.103.01/NVIDIA-Linux-x86_64-470.103.01.run). Al utilizar este método, debe vincular una EIP a cada nodo de la GPU.
- Método 2: Como se muestra en **Figura 14-10**, haga clic en **AGREE & DOWNLOAD** para descargar el controlador. A continuación, cargue el controlador en OBS y registre el URL de OBS. Al utilizar este método, no es necesario vincular una EIP a los nodos de la GPU.

**Figura 14-10** Obtención del enlace



----Fin

## Obtención del enlace del conductor de OBS

**Paso 1** Cargue el controlador a OBS y establezca el archivo del controlador en lectura pública. Para obtener más información, consulte [Carga de un archivo](#).

### NOTA

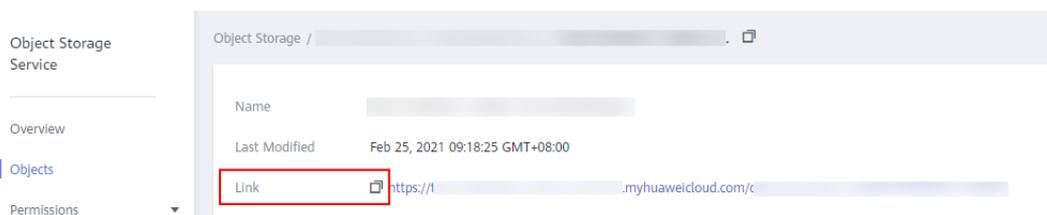
Cuando se reinicia el nodo, el controlador se descargará e instalará de nuevo. Asegúrese de que el enlace de bucket de OBS del controlador es válido.

**Paso 2** En el panel de navegación de la consola OBS, seleccione **Object Storage**.

**Paso 3** En la lista de bucket, haga clic en un nombre de bucket y, a continuación, se mostrará la página **Overview** del bucket.

**Paso 4** En el panel de navegación, elija **Objects**.

**Paso 5** Seleccione el objeto de destino y copie el enlace del controlador en la página de detalles del objeto.



----Fin

## Instalación del servicio de nvidia-fabricmanager

Las GPU A100 y A800 son compatibles con NvLink y NvSwitch. Si utiliza un nodo con varias GPU, debe instalar el servicio nvidia-fabricmanager correspondiente a la versión del controlador para habilitar la interconexión entre GPU. De lo contrario, es posible que los pods de GPU no se utilicen.

Esta sección utiliza el controlador 470.103.01 como ejemplo. Puede realizar los siguientes pasos para instalar el controlador. Reemplace la versión del controlador según sea necesario.

**Paso 1** Inicie sesión en el nodo de la GPU de destino. Una EIP debe estar enlazada al nodo para descargar el servicio nvidia-fabricmanager.

**Paso 2** Instale el servicio nvidia-fabricmanager correspondiente a la versión del controlador. Puede descargar el paquete de instalación correspondiente a su sistema operativo y la versión del controlador desde el [sitio web oficial](#).

- **CentOS**

Tome CentOS 7 como ejemplo:

```
driver_version=470.103.01
wget https://developer.download.nvidia.cn/compute/cuda/repos/rhel7/x86_64/
cuda-drivers-fabricmanager-${driver_version}-1.x86_64.rpm
rpm -ivh nvidia-fabric-manager-${driver_version}-1.x86_64.rpm
```

- **Otros SO como Ubuntu**

Tome Ubuntu 18.04 como ejemplo:

```
driver_version=470.103.01
driver_version_main=$(echo $driver_version | awk -F '.' '{print $1}')
wget https://developer.download.nvidia.cn/compute/cuda/repos/ubuntu1804/
x86_64/nvidia-fabricmanager-${driver_version_main}_$
{driver_version}-1_amd64.deb
dpkg -i nvidia-fabricmanager-${driver_version_main}_$
{driver_version}-1_amd64.deb
```

**Paso 3** Inicie el servicio nvidia-fabricmanager.

```
systemctl enable nvidia-fabricmanager
systemctl start nvidia-fabricmanager
```

**Paso 4** Ejecute el siguiente comando para comprobar el estado del servicio nvidia-fabricmanager:

```
systemctl status nvidia-fabricmanager
```

----Fin

## Enlaces útiles

- [¿Cómo soluciono problemas de gpu-beta y de controladores de GPU?](#)
- [¿Qué debo hacer si ocurren excepciones de nodo de GPU?](#)
- [Programación de GPU](#)

## Historial de cambios

**Tabla 14-25** Versiones de complementos de CCE

Versión del complemento	Versión de clúster admitida
1.2.28	/v1.(19 21 23 25).*/
1.2.24	/v1.(19 21 23 25).*/
1.2.20	/v1.(19 21 23 25).*/
1.2.17	/v1.(15 17 19 21 23).*/
1.2.15	/v1.(15 17 19 21 23).*/
1.2.11	/v1.(15 17 19 21).*/
1.2.10	/v1.(15 17 19 21).*/
1.2.9	/v1.(15 17 19 21).*/
1.2.2	/v1.(15 17 19).*/
1.2.1	/v1.(15 17 19).*/
1.1.13	/v1.(13 15 17).*/

Versión del complemento	Versión de clúster admitida
1.1.11	/v1.(15 17).*/

## 14.13 huawei-npu

### Presentación

Huawei-npu es un complemento de gestión para dispositivos de Huawei NPU en contenedores.

Después de instalar este complemento, puede crear nodos acelerados por Ascend para procesar de forma rápida y eficiente la inferencia y el reconocimiento de imágenes.

### Notas y restricciones

- Si se utilizan nodos acelerados por Ascend en un clúster, se debe instalar el complemento huawei-npu.
- Después de migrar un nodo acelerado por AI, el nodo se restablecerá. Es necesario volver a instalar manualmente el controlador de NPU.

### Instalación del complemento

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster. Elija **Add-ons** en el panel de navegación, localice **huawei-npu** a la derecha y haga clic en **Install**.

**Paso 2** Establezca los parámetros de NPU. El complemento utiliza los siguientes parámetros de forma predeterminada. La configuración predeterminada de NPU proporcionada por el complemento puede satisfacer la mayoría de los escenarios y no requiere cambios.

```
{
  "check_frequency_failed_threshold": 100,
  "check_frequency_fall_times": 3,
  "check_frequency_gate": false,
  "check_frequency_recover_threshold": 100,
  "check_frequency_rise_times": 2,
  "container_path": "/usr/local/HiAI_unused",
  "host_path": "/usr/local/HiAI_unused"
}
```

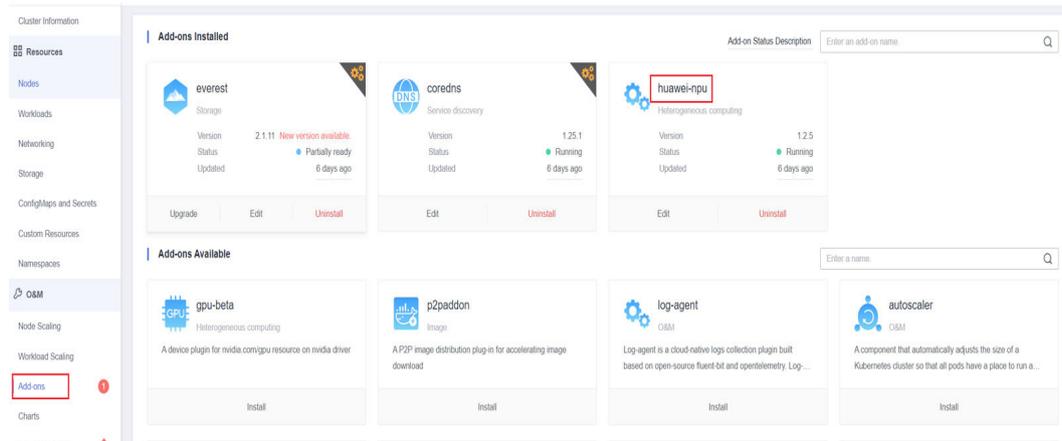
**Paso 3** Haga clic en **Install**.

----Fin

### Cómo comprobar si el controlador de NPU ha sido instalado en un nodo

Después de asegurarse de que el controlador está instalado correctamente, debe reiniciar el nodo para que el controlador surta efecto. De lo contrario, el controlador no puede surtir efecto y los recursos de NPU no están disponibles. Para comprobar si el controlador está instalado, realice las siguientes operaciones:

**Paso 1** En la página **Add-ons** del clúster, haga clic en **huawei-npu** para ir a su lista de pods.



**Paso 2** El estado del pod `npu-driver-installer` es **Running**.

Pods

Pod Name	Status	Names...	Pod IP	Node	Resta...	CPU Request/Limit/Usa	Memory Request/Limit/Usa	Created	Operation
huawei-npu-device	Running	kube-system	192.168.20.59	192.168.20.59	3	0.1 Cores 0.1 Cores 0%	200 MiB 200 MiB 0%	6 days ago	Monitor View Events More
npu-driver-installer	Running	kube-system	192.168.20.59	192.168.20.59	3	0.2 Cores 1 Cores 0%	500 MiB 4 GiB 0%	6 days ago	Monitor View Events More

### NOTA

Si el nodo se reinicia antes de instalar el controlador de NPU, la instalación del controlador puede fallar y se muestra un mensaje en la página **Nodes** del clúster que indica que el controlador Ascend no está listo. En este caso, debe desinstalar el controlador de NPU del nodo y reiniciar el pod `npu-driver-installer` para reinstalar el controlador de NPU. Después de confirmar que el controlador está instalado, reinicie el nodo. Para obtener más información sobre cómo desinstalar el controlador, consulte [Desinstalación del controlador de NPU](#).

----Fin

## Desinstalación del controlador de NPU

Inicie sesión en el nodo, obtenga los registros de operación del controlador en el archivo `/var/log/ascend_seclog/operation.log` y busque el paquete de ejecución del controlador utilizado en la última instalación. Si el archivo log no existe, el controlador se instala utilizando el paquete combinado de controladores `npu_x86_latest.run` o `npu_arm_latest.run`. Después de encontrar el paquete de instalación del controlador, ejecute el comando `bash {run package name} --uninstall` para desinstalar el controlador y reiniciar el nodo como se le solicite.

**Paso 1** Inicie sesión en el nodo donde se debe desinstalar el controlador de NPU y encuentre el archivo `/var/log/ascend_seclog/operation.log`.

**Paso 2** Si se encuentra el archivo `/var/log/ascend_seclog/operation.log`, vea el log de instalación del controlador para encontrar su registro.

```
[root@00379955-w-a1ls-e25 ~]# ll /var/log/ascend_seclog/operation.log
-rw-r----- 1 root root 285 Dec 1 20:00 /var/log/ascend_seclog/operation.log
[root@00379955-w-a1ls-e25 ~]# cat /var/log/ascend_seclog/operation.log
Install SUGGESTION root 2022-12-01 19:53:47 127.0.0.1 Ascend310-hdk-npu-driver 6.0.rc1 linux-x86-64.run success install_type=full; cmdlist=--quiet --full.
```

Si no se encuentra el archivo `/var/log/ascend_seclog/operation.log`, el controlador se puede instalar utilizando el paquete combinado de controladores `npu_x86_latest.run` o `npu_arm_latest.run`. Puede confirmar esto comprobando si el directorio `/usr/local/HIAI/driver/` existe.

 **NOTA**

El paquete combinado del controlador de NPU se almacena en el directorio `/root/d310_driver` y otros paquetes de instalación de controladores se almacenan en el directorio `/root/npu-drivers`.

**Paso 3** Después de encontrar el paquete de instalación del controlador, ejecute el comando `bash {run package path} --uninstall` para desinstalar el controlador. A continuación se utiliza `Ascend310-hdk-npu-driver_6.0.rc1_linux-x86-64.run` como ejemplo:

```
bash /root/npu-drivers/Ascend310-hdk-npu-driver_6.0.rc1_linux-x86-64.run --uninstall
```

```
[root@y00379955-w-ails-e25 npu-drivers]# ./Ascend310-hdk-npu-driver_6.0.rc1_linux-x86-64.run --uninstall
Verifying archive integrity... 100% SHA256 checksums are OK. All good.
Uncompressing ASCEND DRIVER RUN PACKAGE 100%
[Driver] [2022-12-01 19:59:53] [INFO]Start time: 2022-12-01 19:59:53
[Driver] [2022-12-01 19:59:53] [INFO]LogFile: /var/log/ascend_seclog/ascend_install.log
[Driver] [2022-12-01 19:59:53] [INFO]OperationLogFile: /var/log/ascend_seclog/operation.log
[Driver] [2022-12-01 19:59:53] [INFO]base version is 22.0.3.

[Driver] [2022-12-01 20:00:04] [INFO]Driver package uninstalled successfully! Reboot needed for uninstallation to take effect!
[Driver] [2022-12-01 20:00:04] [INFO]End time: 2022-12-01 20:00:04
```

**Paso 4** Reinicie el nodo como se le solicite. (La instalación y la desinstalación del controlador de NPU actual solo tienen efecto después de reiniciar el nodo.)

----Fin

## Historial de cambios

**Tabla 14-26** Versiones de complementos de CCE

Versión del complemento	Versión de clúster admitida
1.2.5	/v1.(19 21 23 25).*/
1.2.4	/v1.(19 21 23 25).*/
1.2.2	/v1.(19 21 23).*/
1.2.1	/v1.(19 21 23).*/
1.1.8	/v1.(15 17 19 21).*/
1.1.2	/v1.(15 17 19).*/
1.1.1	/v1.(15 17 19).*/
1.1.0	/v1.(17 19).*/
1.0.8	/v1.(13 15 17).*/
1.0.6	/v1.(13 15 17).*/
1.0.5	/v1.(13 15 17).*/
1.0.3	/v1.(13 15 17).*/

## 14.14 Volcano

### Presentación

Volcano es una plataforma de procesamiento por lotes basada en Kubernetes. Proporciona una serie de características requeridas por aprendizaje automático, aprendizaje profundo, bioinformática, genómica y otras aplicaciones de big data, como un poderoso complemento a las capacidades de Kubernetes.

Volcano proporciona capacidades informáticas de alto rendimiento de propósito general, como motor de programación de trabajos, gestión de chips heterogéneos y gestión de ejecución de trabajos, sirviendo a los usuarios finales con marcos informáticos para diferentes industrias, como IA, big data, secuenciación de genes y renderizado. (Volcano ha sido de código abierto en el GitHub.)

Volcano proporciona programación de trabajos, gestión de trabajos y gestión de colas para las aplicaciones informáticas. Sus principales características son las siguientes:

- En Kubernetes de contenedores se pueden ejecutar diversos marcos de cómputo, como TensorFlow, MPI y Spark. Se proporcionan las API comunes para trabajos de cómputo por lotes con CRD, varios complementos y gestión avanzada del ciclo de vida de trabajos.
- Las capacidades avanzadas de programación se proporcionan para el cómputo por lotes y los escenarios de cómputo de alto rendimiento, incluidos la programación de grupos, la programación de prioridades preventivas, el empaquetado, la reserva de recursos y la topología de tareas.
- Las colas se pueden gestionar eficazmente para la programación de trabajos. Se admiten las capacidades complejas de programación de trabajos, como la prioridad de cola y las colas de varios niveles.

Comunidad de código abierto: <https://github.com/volcano-sh/volcano>

### Instalación del complemento

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster. Elija **Add-ons** en el panel de navegación, localice **volcano** a la derecha y haga clic en **Install**.

**Paso 2** Seleccione **Standalone**, **Custom** o **HA** para **Add-on Specifications**.

Si selecciona **Custom**, los valores recomendados para **volcano-controller** y **volcano-scheduler** son los siguientes:

- Si el número de nodos es inferior a 100, conserve la configuración predeterminada. Es decir, el valor de petición de la CPU es **500m** y el valor límite es **2000m**. El valor de solicitud de memoria es **500Mi** y el valor límite es **2000Mi**.
- Si el número de nodos es mayor que 100, aumente el valor de solicitud de CPU en **500m** y el valor de solicitud de memoria en **1000Mi** cada vez que se agreguen 100 nodos (pods de 10,000). Se recomienda aumentar el valor límite de CPU en **1500m** y el límite de memoria en **1000Mi**.

**Tabla 14-27** Valores recomendados para volcano-controller and volcano-scheduler

Número de nodos/pods	Solicitud de CPU(m)	Límite de CPU(m)	Solicitud de memoria(Mi)	Límite de memoria(Mi)
50/5000	500	2000	500	2000
100/10,000	1000	2500	1500	2500
200/20,000	1500	3000	2500	3500
300/30,000	2000	3500	3500	4500
400/40,000	2500	4000	4500	5500

**Paso 3** Configure los parámetros del planificador de volcano predeterminado. Para obtener más información, véase [Tabla 14-28](#).

```

colocation_enable: ''
default_scheduler_conf:
  actions: 'allocate, backfill'
  tiers:
    - plugins:
      - name: 'priority'
      - name: 'gang'
      - name: 'conformance'
    - plugins:
      - name: 'drf'
      - name: 'predicates'
      - name: 'nodeorder'
    - plugins:
      - name: 'cce-gpu-topology-predicate'
      - name: 'cce-gpu-topology-priority'
      - name: 'cce-gpu'
    - plugins:
      - name: 'nodelocalvolume'
      - name: 'nodeemptydirvolume'
      - name: 'nodeCSIscheduling'
      - name: 'networkresource'
    
```

**Tabla 14-28** Complementos de Volcano

Complemento	Función	Descripción	Demostración
binpack	Programa los pods en nodos con una alta utilización de recursos para reducir los fragmentos de recursos.	<ul style="list-style-type: none"> <li>● <b>binpack.weight</b>: ponderación del complemento binpack.</li> <li>● <b>binpack.cpu</b>: relación de recursos de CPU a todos los recursos. El valor predeterminado es <b>1</b>.</li> <li>● <b>binpack.memory</b>: relación entre los recursos de memoria y todos los recursos. El valor predeterminado es <b>1</b>.</li> <li>● <b>binpack.resources</b>: tipo de recurso.</li> </ul>	<pre>- plugins:   - name: binpack     arguments:       binpack.weight: 10       binpack.cpu: 1       binpack.memory: 1  binpack.resources:   nvidia.com/gpu,   example.com/foo  binpack.resources.nvidia.com/gpu: 2  binpack.resources.example.com/foo: 3</pre>
conformance	El complemento de conformidad considera que las tareas en el espacio de nombres <b>kube-system</b> tienen una prioridad más alta. Estas tareas no serán prevaricadas.	-	-
gang	El complemento gang considera un grupo de pods como un todo para asignar recursos.	-	-
priority	El complemento priority programa los pods en función de la prioridad de carga de trabajo personalizada.	-	-

Completo	Función	Descripción	Demostración
overcommit	Los recursos de un clúster se programan después de acumularse en un determinado múltiplo para mejorar la eficiencia de la cola de carga de trabajo. Si todas las cargas de trabajo son Deployments, quite este complemento o establezca el factor de aumento en <b>2.0</b> .	<b>overcommit-factor:</b> Factor de aumento. El valor predeterminado es <b>1.2</b> .	<pre data-bbox="1123 327 1426 456">- plugins:   - name: overcommit     arguments:       overcommit-factor: 2.0</pre>
drf	Programa los recursos basados en los recursos dominantes del grupo de contenedor. Los recursos de dominantes más pequeños se seleccionarían para la programación de prioridades.	-	-
predicates	Determina si una tarea está vinculada a un nodo mediante una serie de algoritmos de evaluación, como afinidad de nodo/pod, tolerancia a la contaminación, repetición de puertos de nodo, límites de volumen y coincidencia de zona de volumen.	-	-

Complemento	Función	Descripción	Demostración
nodeorder	El complemento nodeorder puntua todos los nodos para una tarea usando una serie de algoritmos de puntuación.	<ul style="list-style-type: none"> <li>● <b>nodeaffinity.weight:</b> Los pods se programan en función de la afinidad del nodo. El valor predeterminado es <b>1</b>.</li> <li>● <b>podaffinity.weight:</b> Los pods se programan en función de la afinidad de pods. El valor predeterminado es <b>1</b>.</li> <li>● <b>leastrequested.weight:</b> Los pods se programan para el nodo con los recursos menos solicitados. El valor predeterminado es <b>1</b>.</li> <li>● <b>balancedresource.weight:</b> Los pods se programan en el nodo con el recurso equilibrado. El valor predeterminado es <b>1</b>.</li> <li>● <b>mostrequested.weight:</b> Los pods se programan para el nodo con los recursos más solicitados. Defaults to <b>0</b>.</li> <li>● <b>tainttoleration.weight:</b> Los pods se programan en el nodo con una alta tolerancia a la contaminación. El valor predeterminado es <b>1</b>.</li> <li>● <b>imagelocality.weight:</b> Los pods se programan en el nodo donde existen las imágenes requeridas. El valor predeterminado es <b>1</b>.</li> <li>● <b>selectorspread.weight:</b> Los pods se programan uniformemente para diferentes nodos. El valor predeterminado es <b>0</b>.</li> <li>● <b>volumebinding.weight:</b> Los pods se programan para el nodo con la</li> </ul>	<pre> - plugins:   - name: nodeorder     arguments:       leastrequested.weight: 1       mostrequested.weight: 0       nodeaffinity.weight: 1       podaffinity.weight: 1       balancedresource.weight: 1       tainttoleration.weight: 1       imagelocality.weight: 1       volumebinding.weight: 1       podtopologyspread.weight: 2                     </pre>

Complemento	Función	Descripción	Demostración
		<p>política de enlace retardado de PV local. El valor predeterminado es <b>1</b>.</p> <ul style="list-style-type: none"> <li>● <b>podtopologyspread.weight</b>: Los pods se programan en función de la topología del pod. El valor predeterminado es <b>2</b>.</li> </ul>	
cce-gpu-topology - predicate	Algoritmo de preselección de programación de topología de GPU	-	-
cce-gpu-topology -priority	Algoritmo de prioridad de programación de topología de GPU	-	-
cce-gpu	Funciona con el complemento gpu de CCE para admitir la asignación de recursos de GPU y la configuración decimal de GPU.	-	-
numaaware	Programación de topología NUMA	<b>weight</b> : Ponderación del complemento numa-aware.	-
networkresource	El nodo de requisitos de ENI puede preseleccionarse y filtrarse. Los parámetros son transferidos por CCE y no necesitan ser configurados manualmente.	<b>NetworkType</b> : tipo de red ( <b>eni</b> o <b>vpc-router</b> ).	-
nodelocalvolume	Filtra los nodos que no cumplen los requisitos de volumen local.	-	-

Completo	Función	Descripción	Demostración
nodeemptydirvolume	Filtra los nodos que no cumplen los requisitos de emptyDir.	-	-
nodeCSIScheduling	Filtra los nodos que tienen siempre excepciones de componente.	-	-

**Paso 4** Haga clic en **Install**.

---Fin

## Modificación de las configuraciones de volcano-scheduler con la consola

Volcano le permite configurar el planificador durante la instalación, actualización y edición. La configuración se sincronizará con volcano-scheduler-configmap.

Esta sección describe cómo configurar volcano-scheduler.

### NOTA

Solo Volcano de v1.7.1 y posteriores soportan esta función. En la nueva página del complemento, opciones como **plugins.eas\_service** y **resource\_exporter\_enable** se sustituyen por **default\_scheduler\_conf**.

Inicie sesión en la consola de CCE y acceda a la consola del clúster. Elija **Add-ons** en el panel de navegación. A la derecha de la página, localice **volcano** y haga clic en **Install** o **Upgrade**. En el área **Parameters**, configure los parámetros de volcano-scheduler.

- Uso de **resource\_exporter**:

```
{
  "ca_cert": "",
  "default_scheduler_conf": {
    "actions": "allocate, backfill",
    "tiers": [
      {
        "plugins": [
          {
            "name": "priority"
          },
          {
            "name": "gang"
          },
          {
            "name": "conformance"
          }
        ]
      },
      {
        "plugins": [
          {
            "name": "drf"
          }
        ]
      }
    ]
  }
}
```

```

        "name": "predicates"
    },
    {
        "name": "nodeorder"
    }
]
},
{
    "plugins": [
        {
            "name": "cce-gpu-topology-predicate"
        },
        {
            "name": "cce-gpu-topology-priority"
        },
        {
            "name": "cce-gpu"
        },
        {
            "name": "numa-aware" # add this also enable
resource_exporter
        }
    ]
},
{
    "plugins": [
        {
            "name": "nodelocalvolume"
        },
        {
            "name": "nodeemptydirvolume"
        },
        {
            "name": "nodeCSIscheduling"
        },
        {
            "name": "networkresource"
        }
    ]
}
},
"server_cert": "",
"server_key": ""
}
    
```

Después de que esta función esté habilitada, puede usar las funciones del complemento numa-aware y resource\_exporter al mismo tiempo.

- **Uso de eas\_service:**

```

{
    "ca_cert": "",
    "default_scheduler_conf": {
        "actions": "allocate, backfill",
        "tiers": [
            {
                "plugins": [
                    {
                        "name": "priority"
                    },
                    {
                        "name": "gang"
                    },
                    {
                        "name": "conformance"
                    }
                ]
            }
        ],
    },
    {
        "plugins": [
    
```

```

        {
            "name": "drf"
        },
        {
            "name": "predicates"
        },
        {
            "name": "nodeorder"
        }
    ]
},
{
    "plugins": [
        {
            "name": "cce-gpu-topology-predicate"
        },
        {
            "name": "cce-gpu-topology-priority"
        },
        {
            "name": "cce-gpu"
        },
        {
            "name": "eas",
            "custom": {
                "availability_zone_id": "",
                "driver_id": "",
                "endpoint": "",
                "flavor_id": "",
                "network_type": "",
                "network_virtual_subnet_id": "",
                "pool_id": "",
                "project_id": "",
                "secret_name": "eas-service-secret"
            }
        }
    ]
},
{
    "plugins": [
        {
            "name": "nodelocalvolume"
        },
        {
            "name": "nodeemptydirvolume"
        },
        {
            "name": "nodeCSIScheduling"
        },
        {
            "name": "networkresource"
        }
    ]
}
    ]
},
"server_cert": "",
"server_key": ""
}

```

● Using **ief**:

```

{
    "ca_cert": "",
    "default_scheduler_conf": {
        "actions": "allocate, backfill",
        "tiers": [
            {
                "plugins": [
                    {
                        "name": "priority"
                    }
                ]
            }
        ]
    }
}

```

```

        },
        {
            "name": "gang"
        },
        {
            "name": "conformance"
        }
    ]
},
{
    "plugins": [
        {
            "name": "drf"
        },
        {
            "name": "predicates"
        },
        {
            "name": "nodeorder"
        }
    ]
},
{
    "plugins": [
        {
            "name": "cce-gpu-topology-predicate"
        },
        {
            "name": "cce-gpu-topology-priority"
        },
        {
            "name": "cce-gpu"
        },
        {
            "name": "ief",
            "enableBestNode": true
        }
    ]
},
{
    "plugins": [
        {
            "name": "nodelocalvolume"
        },
        {
            "name": "nodeemptydirvolume"
        },
        {
            "name": "nodeCSIScheduling"
        },
        {
            "name": "networkresource"
        }
    ]
}
    ],
    "server_cert": "",
    "server_key": ""
}
    
```

## Retención de las configuraciones originales de volcano-scheduler-configmap

Si desea utilizar la configuración original después de actualizar el complemento, realice los siguientes pasos:

- Paso 1** Compruebe y haga una copia de respaldo de la configuración original de volcano-scheduler-configmap.

**Por ejemplo:**

```
# kubectl edit cm volcano-scheduler-configmap -n kube-system
apiVersion: v1
data:
  default-scheduler.conf: |-
    actions: "enqueue, allocate, backfill"
    tiers:
    - plugins:
      - name: priority
      - name: gang
      - name: conformance
    - plugins:
      - name: drf
      - name: predicates
      - name: nodeorder
      - name: binpack
      arguments:
        binpack.cpu: 100
        binpack.weight: 10
        binpack.resources: nvidia.com/gpu
        binpack.resources.nvidia.com/gpu: 10000
    - plugins:
      - name: cce-gpu-topology-predicate
      - name: cce-gpu-topology-priority
      - name: cce-gpu
    - plugins:
      - name: nodelocalvolume
      - name: nodeemptydirvolume
      - name: nodeCSIScheduling
      - name: networkresource
```

**Paso 2** Introduzca el contenido personalizado en el área **Parameters** de la consola.

```
{
  "ca_cert": "",
  "default_scheduler_conf": {
    "actions": "enqueue, allocate, backfill",
    "tiers": [
      {
        "plugins": [
          {
            "name": "priority"
          },
          {
            "name": "gang"
          },
          {
            "name": "conformance"
          }
        ]
      },
      {
        "plugins": [
          {
            "name": "drf"
          },
          {
            "name": "predicates"
          },
          {
            "name": "nodeorder"
          },
          {
            "name": "binpack",
            "arguments": {
              "binpack.cpu": 100,
              "binpack.weight": 10,
              "binpack.resources": "nvidia.com/gpu",
              "binpack.resources.nvidia.com/gpu": 10000
            }
          }
        ]
      }
    ]
  }
}
```

```
    }
  ],
},
{
  "plugins": [
    {
      "name": "cce-gpu-topology-predicate"
    },
    {
      "name": "cce-gpu-topology-priority"
    },
    {
      "name": "cce-gpu"
    }
  ]
},
{
  "plugins": [
    {
      "name": "nodelocalvolume"
    },
    {
      "name": "nodeemptydirvolume"
    },
    {
      "name": "nodeCSIScheduling"
    },
    {
      "name": "networkresource"
    }
  ]
}
]
},
"server_cert": "",
"server_key": ""
}
```

#### **NOTA**

Cuando se utiliza esta función, se sobrescribirá el contenido original en volcano-scheduler-configmap. Por lo tanto, debe verificar si volcano-scheduler-configmap ha sido modificado durante la actualización. Si es así, sincronice la modificación en la página de actualización.

---Fin

## Operaciones relacionadas

- [Despliegue híbrido de trabajos en línea y fuera de línea](#)
- [Programación de afinidad de NUMA](#)

## Historial de cambios

---

### AVISO

Se recomienda actualizar Volcano a la última versión que coincida con el clúster.

---

**Tabla 14-29** Asignación de versión de clúster

Versión del clúster	Versión del complemento
v1.25	1.7.1 y 1.7.2
v1.23	1.7.1 y 1.7.2
v1.21	1.7.1 y 1.7.2
v1.19.16	1.3.7, 1.3.10, 1.4.5, 1.7.1 y 1.7.2
v1.19	1.3.7, 1.3.10 y 1.4.5
v1.17 (Fin de mantenimiento)	1.3.7, 1.3.10 y 1.4.5
v1.15 (Fin de mantenimiento)	1.3.7, 1.3.10 y 1.4.5

**Tabla 14-30** Versiones de complementos de CCE

Versión del complemento	Versión de clúster admitida	Característica actualizada
1.7.2	/v1.19.16.* v1.21.* v1.23.* v1.25.*	<ul style="list-style-type: none"> <li>● Compatible con Kubernetes 1.25.</li> <li>● Mejorada la programación del volcano.</li> </ul>
1.7.1	/v1.19.16.* v1.21.* v1.23.* v1.25.*	Compatible con Kubernetes 1.25.
1.6.5	/v1.19.* v1.21.* v1.23.*	<ul style="list-style-type: none"> <li>● Sirve como el programador predeterminado de CCE.</li> <li>● Admitida la programación unificada en despliegues híbridos.</li> </ul>
1.4.5	/v1.17.* v1.19.* v1.21.*	<ul style="list-style-type: none"> <li>● Cambiado el modo de despliegue del volcano-scheduler desde <b>statefulset</b> hasta <b>deployment</b>. Se ha corregido el problema de que los pods no se pueden migrar automáticamente cuando el nodo es anormal.</li> </ul>
1.4.2	/v1.15.* v1.17.* v1.19.* v1.21.*	<ul style="list-style-type: none"> <li>● Resuelto el problema de que la asignación entre las GPU falla.</li> <li>● Compatible con la API de EAS actualizada.</li> </ul>
1.3.3	/v1.15.* v1.17.* v1.19.* v1.21.*	<ul style="list-style-type: none"> <li>● Corregido el problema de bloqueo del planificador causado por las excepciones de GPU y el problema de error de admisión para contenedores de inicio privilegiado.</li> </ul>

Versión del complemento	Versión de clúster admitida	Característica actualizada
1.3.1	/v1.15.* v1.17.* v1.19.*	<ul style="list-style-type: none"> <li>● Actualizado el firmware de la tarjeta controladora RAID a la versión más reciente.</li> <li>● Compatible con Kubernetes 1.19.</li> <li>● Agregado el complemento numa-aware.</li> <li>● Corregido el problema de ajuste de despliegue en el escenario de colas múltiples.</li> <li>● Ajustado el complemento de algoritmo habilitado de forma predeterminada.</li> </ul>
1.2.5	/v1.15.* v1.17.* v1.19.*	<ul style="list-style-type: none"> <li>● Corregido el problema de OutOfcpu en algunos escenarios.</li> <li>● Corregido el problema de que los pods no se pueden programar cuando se configuran algunas capacidades para una cola.</li> <li>● El tiempo de log del componente de volcano es consistente con el tiempo del sistema.</li> <li>● Corregido el problema de preferencia entre varias colas.</li> <li>● Solucionado el problema de que el resultado del complemento IoAware no cumple con la expectativa en algunos escenarios extremos.</li> <li>● Compatibles con los clústeres híbridos.</li> </ul>

Versión del complemento	Versión de clúster admitida	Característica actualizada
1.2.3	/v1.15.* v1.17.* v1.19.*	<ul style="list-style-type: none"> <li>● Corregido el problema de OOM de la tarea de entrenamiento causado por la precisión insuficiente.</li> <li>● Corregido el problema de programación de GPU en CCE 1.15 y versiones posteriores. No se admite la actualización continua de las versiones de CCE durante la distribución de tareas.</li> <li>● Corregido el problema por el que se desconocía el estado de la cola en ciertos escenarios.</li> <li>● Corregido el problema por el que se producía un pánico cuando se montaba un PVC en un trabajo en un escenario específico.</li> <li>● Corregido el problema de que los decimales no se podían configurar para los trabajos de GPU.</li> <li>● Agregado el complemento ioaware.</li> <li>● Agregado el controlador de anillo.</li> </ul>

## 14.15 dew-provider

### Presentación

El complemento de dew-provider se utiliza para interconectar con [Data Encryption Workshop \(DEW\)](#), que le permite montar secretos almacenados fuera de un clúster (es decir, DEW para almacenar información confidencial) en los pods. De esta manera, la información sensible puede desacoplarse del entorno de agrupamiento, evitando la fuga de información causada por la configuración de codificación dura del programa o de texto plano.

### Notas y restricciones

- DEW incluye Key Management Service (KMS), Cloud Secret Management Service (CSMS) y Key Pair Service (KPS). Actualmente, el complemento de dew-provider solo puede interconectarse con CSMS.
- El complemento de dew-provider solo se puede instalar en clústeres v1.19 o posterior.
- El complemento de dew-provider se puede instalar en clústeres de CCE y en clústeres de CCE Turbo.
- Se puede crear un máximo de 500 objetos de SecretProviderClass.
- Cuando se desinstala el complemento, los recursos de CRD relacionados se eliminan en consecuencia. Incluso si se reinstala el complemento, el objeto de SecretProviderClass

original no está disponible. Si desea utilizar los recursos originales de SecretProviderClass después de desinstalar y reinstalar el complemento, debe crearlos manualmente de nuevo.

## Cómo funciona el complemento

### Principios básicos

El complemento de dew-provider consiste en secrets-store-csi-driver y dew-provider, que se despliegan como DaemonSets.

- El componente secrets-store-csi-driver es responsable de mantener dos CRD: SecretProviderClass (SPC) y SecretProviderClassPodStatus (spcPodStatus). **SPC** se utiliza para describir el secreto en el que los usuarios están interesados (como la versión secreta y el nombre). Es creado por los usuarios y será referenciado en los pods. **spcPodStatus** se utiliza para rastrear las relaciones de unión entre los pods y secretos. Es creado automáticamente por csi-driver y no requiere operación manual. Un pod corresponde a un spcPodStatus. Después de iniciar un pod, se genera un spcPodStatus para el pod. Cuando finaliza el ciclo de vida del pod, el spcPodStatus se elimina en consecuencia.
- El componente de dew-provider obtiene los secretos especificados de CSMS y los monta en los pods.

### Funciones

- Montaje básico: Después de instalar el complemento de dew-provider, puede crear un objeto de SecretProviderClass y declarar y hacer referencia al volumen en un pod. Cuando se inicia el pod, el secreto declarado en el objeto SecretProviderClass se monta en el pod.
- Rotación programada: Después de que un pod se ejecute correctamente, si se actualiza el secreto declarado en el objeto SPC y almacenado en CSMS, los últimos valores secretos se pueden actualizar en el pod con la rotación programada. Cuando se utiliza esta capacidad, debe establecer la versión secreta en **latest**.
- Conocimiento en tiempo real de los cambios de SPC: Después de que un pod se ejecute correctamente, si un usuario modifica el secreto declarado en el objeto de SPC (por ejemplo, se agrega un secreto o se cambia el número de versión), el complemento puede detectar el cambio en tiempo real y actualizar el secreto al pod.

## Instalación del complemento

- Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster. Elija **Add-ons** en el panel de navegación, localice **dew-provider** a la derecha y haga clic en **Install**.
- Paso 2** En la página **Install Add-on**, seleccione el clúster donde se va a instalar el complemento y configure los parámetros en el área **Parameters** como se indica en la siguiente tabla.

Parámetro	Descripción
rotation_poll_interval	Intervalo de rotación, en unidad de m (en lugar de min). El intervalo de rotación indica el intervalo para enviar una solicitud al CSMS y obtener el último secreto. El intervalo adecuado es [1m, 1440m]. El valor predeterminado es <b>2m</b> .

**Paso 3** Haga clic en **Install**.

Después de instalar el complemento, seleccione el clúster y haga clic en **Add-ons** en el panel de navegación. En la página mostrada, vea el complemento en el área **Add-ons Installed**.

---Fin

## Uso de complementos

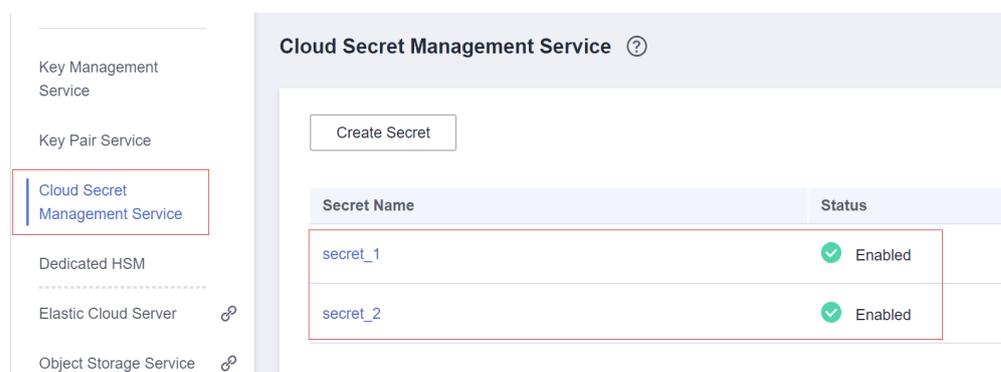
**Paso 1** Crear un ServiceAccount.

1. Cree un objeto de ServiceAccount, **que declara los nombres secretos que pueden ser utilizados por los servicios. Si un usuario hace referencia a un secreto que no se declara aquí, el montaje fallará. Como resultado, el pod no puede funcionar.**

Cree el archivo **serviceaccount.yaml** basado en la siguiente plantilla y declare los nombres secretos que pueden usar los servicios en el campo **cce.io/dew-resource**. Aquí, se declaran **secret\_1** y **secret\_2** indicando que se permite al servicio hacer referencia a dos secretos. En operaciones posteriores, si el usuario hace referencia a **secret\_3** en el servicio, la verificación falla. Como resultado, el secreto no se puede montar y el pod no puede funcionar.

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: nginx-spc-sa
  annotations:
    cce.io/dew-resource: "[\"secret_1\", \"secret_2\"]" #secrets that allow
    pod to use
```

Asegúrese de que los secretos declarados aquí existen en CSCM, como se muestra en la siguiente figura. De lo contrario, incluso si la verificación tiene éxito, se produce un error cuando el secreto correspondiente se obtiene de CSCM. Como resultado, la cápsula no puede funcionar correctamente.



2. Ejecute el siguiente comando para crear el ServiceAccount:

**kubectl apply -f serviceaccount.yaml**

3. Compruebe si el objeto ServiceAccount se ha creado correctamente.

```
$ kubectl get sa
NAME          SECRETS  AGE
default      1        18d # This is the default ServiceAccount object
of the system.
nginx-spc-sa  1        19s # This is the newly created ServiceAccount
object.
```

Se ha creado un objeto de ServiceAccount denominado **nginx-spc-sa**. Se hará referencia a este objeto en los pods.

**Paso 2** Crear un SecretProviderClass.

1. El objeto de SecretProviderClass se utiliza para describir la información secreta (como la versión y el nombre) que interesan a los usuarios. Es creado por los usuarios y será referenciado en los pods.

Cree el archivo **secretproviderclass.yaml** utilizando la plantilla a continuación. Preste atención al campo **objects** de **parameters** que es un array utilizado para declarar el secreto a montar.

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
  name: spc-test
spec:
  provider: cce # The value is fixed at cce.
  parameters:
    objects: |
      - objectName: "secret_1"
        objectVersion: "v1"
        objectType: "csms"
```

Parámetro	Tipo	Obligatorio	Descripción
objectName	String	Sí	Nombre de la credencial. Si se definen varios objectNames en el mismo SecretProviderClass, los objectNames deben ser únicos. De lo contrario, el montaje falla.
objectAlias	String	No	Nombre de archivo del secreto escrito en el contenedor. Si no se especifica este parámetro, el nombre de archivo del secreto escrito en el contenedor es el valor de <b>objectName</b> de forma predeterminada. Si se especifica este parámetro, el valor debe ser diferente de <b>objectName</b> y de los valores <b>objectAlias</b> y <b>objectName</b> de otros secretos. De lo contrario, el montaje falla.
objectType	String	Sí	Tipo secreto. Actualmente, solo se admite <b>csms</b> . Otros valores no son válidos.

Parámetro	Tipo	Obligatorio	Descripción
objectVersion	String	Sí	Versión secreta. <ul style="list-style-type: none"> <li>– Especifique una versión, por ejemplo, v1.</li> <li>– Utilice la última versión (última). Cuando <b>objectVersion</b> se establece en <b>latest</b>, si se actualiza el secreto correspondiente en CSCM, se actualizará en el pod después de un cierto intervalo (<b>rotation_poll_interval</b>).</li> </ul>

2. Ejecute el siguiente comando para crear un objeto de SecretProviderClass:

**kubectl apply -f secretproviderclass.yaml**

3. Compruebe si se ha creado el objeto de SecretProviderClass.

```
$ kubectl get spc
NAME    AGE
spc-test 20h
```

Se crea un objeto SecretProviderClass denominado **spc-test**. Este objeto será referenciado en pods posteriormente.

### Paso 3 Crear un pod.

A continuación se describe cómo crear una aplicación de Nginx.

1. Defina una carga de trabajo, haga referencia al objeto de ServiceAccount creado en **serviceAccountName** y haga referencia al objeto de SPC creado en **secretProviderClass**, especifique la ruta de montaje del contenedor en **mountPath**. (No especifique directorios especiales como / y **/var/run**. Otherwise, the container may fail to be started.)

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-spc
  labels:
    app: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      serviceAccountName: nginx-spc-sa # Reference the created
      ServiceAccount.
      volumes:
        - name: secrets-store-inline
          csi:
            driver: secrets-store.csi.k8s.io
            readOnly: true
            volumeAttributes:
              secretProviderClass: "spc-test" # Reference the created SPC.
      containers:
        - name: nginx-spc
          image: nginx:alpine
          imagePullPolicy: IfNotPresent
          volumeMounts:
```

```
- name: secrets-store-inline
  mountPath: "/mnt/secrets-store" # Define the mount path of
  secrets in the container.
  readOnly: true
  imagePullSecrets:
  - name: default-secret
```

2. Ejecute **kubectl apply -f despliegue.yaml** para crear un pod.
3. Compruebe si se ha creado el pod.

```
$ kubectl get pod
NAME                                READY   STATUS    RESTARTS   AGE
nginx-spc-67c9d5b594-642np         1/1     Running   0           20s
```

4. Acceda al contenedor y compruebe si el secreto especificado está escrito correctamente. Por ejemplo:

```
$ kubectl exec -ti nginx-spc-67c9d5b594-642np -- /bin/bash
root@nginx-spc-67c9d5b594-642np:/#
root@nginx-spc-67c9d5b594-642np:/# cd /mnt/secrets-store/
root@nginx-spc-67c9d5b594-642np:/mnt/secrets-store#
root@nginx-spc-67c9d5b594-642np:/mnt/secrets-store# ls
secret_1
```

El resultado del comando muestra que `secret_1` declarado en el objeto de SPC se ha escrito en el pod.

Además, puede obtener **spcPodStatus** para comprobar la relación de enlace entre pods y secretos. Por ejemplo:

```
$ kubectl get spcps
NAME                                                                AGE
nginx-spc-67c9d5b594-642np-default-spc-test 103s
$ kubectl get spcps nginx-spc-67c9d5b594-642np-default-spc-test -o yaml
.....
status:
mounted: true
objects: # Mounted secret
- id: secret_1
version: v1
podName: nginx-spc-67c9d5b594-642np # Pod that references the SPC object
secretProviderClassName: spc-test # SPC object
targetPath: /mnt/paas/kubernetes/kubelet/pods/6dd29596-5b78-44fb-9d4c-
a5027c420617/volumes/kubernetes.io~csi/secrets-store-inline/mount
```

----Fin

## Rotación programada

**Como se describió antes**, puede usar este complemento para completar los secretos de montaje, es decir, puede escribir los secretos almacenados en CSMS en un pod.

Para cambiar la versión secreta declarada en el objeto SPC a **latest**, ejecute el siguiente comando:

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
  name: spc-test
spec:
  provider: cce
  parameters:
    objects: |
      - objectName: "secret_1"
        objectVersion: "latest" # change "v1" to "latest"
        objectType: "csms"
```

Después de actualizar el objeto de SPC, el complemento envía periódicamente una solicitud al CSMS para obtener el valor de `secret_1` de la última versión y actualiza el valor al pod que

hace referencia al objeto SPC. El intervalo para que el complemento envíe solicitudes periódicamente se especifica mediante **rotation\_poll\_interval** establecido en [Instalación del complemento](#).

## Detección en tiempo real de cambios de SPC

Los cambios de SPC ya se detectan en tiempo real en [Uso de complementos](#) y [Rotación programada](#). Para una demostración, agregue el secreto **secret\_2** al objeto SPC de la siguiente manera:

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
  name: spc-test
spec:
  provider: cce
  parameters:
    objects: |
      - objectName: "secret_1"
        objectVersion: "latest"
        objectType: "csms"
      - objectName: "secret_2"
        objectVersion: "v1"
        objectType: "csms"
```

Una vez actualizado el objeto de SPC, el nuevo **secret\_2** se monta rápidamente en el pod que hace referencia al objeto de SPC.

## Consulta de logs de componentes

Ver el pod donde se ejecuta el complemento.

```
$ kubectl get pod -n kube-system
NAME                READY   STATUS    RESTARTS   AGE
csi-secrets-store-76tj2    3/3    Running   0           11h
dew-provider-hm5fq        1/1    Running   0           11h
```

Vea los logs de pod del componente de dew-provider.

```
$ kubectl logs dew-provider-hm5fq -n kube-system
...Log information omitted...
...
```

Vea los logs de pod del componente csi-secrets-store. Como el pod del componente csi-secrets-store contiene varios contenedores, debe ejecutar el comando **-c** para especificar un contenedor al ver los logs de pod. El contenedor de secrets-store es el principal contenedor de servicios del complemento y contiene la mayoría de los logs.

```
$ kubectl logs csi-secrets-store-76tj2 -c secrets-store -n kube-system
...Log information omitted...
...
```

## Historial de cambios

**Tabla 14-31** Versiones de complementos de CCE

Versión del complemento	Versión de clúster admitida
1.0.3	/v1.(19 21 23 25).*/
1.0.2	/v1.(19 21 23).*/

Versión del complemento	Versión de clúster admitida
1.0.1	/v1.(19 21).*/

## 14.16 dolphin

### Presentación

dolphin es un complemento para monitorear y gestionar el tráfico de red contenedor. dolphin de la versión actual puede recopilar estadísticas de tráfico de Kata y los contenedores comunes en clústeres de CCE Turbo.

Este complemento recopila cuántos paquetes de IPv4 y bytes se reciben y envían (incluidos los enviados a la red pública). PodSelectors se puede usar para seleccionar backends de monitoreo para admitir múltiples tareas de monitoreo y métricas de monitoreo opcionales. También puede obtener información sobre la etiqueta de los pods. La información de seguimiento se ha adaptado al formato de Prometheus. Puede invocar a la API de Prometheus para ver los datos de monitoreo.

### Restricciones

- Este complemento solo se puede instalar en clústeres de CCE Turbo de la versión 1.19 o posterior. Sus pods solo se pueden desplegar en nodos que ejecuten EulerOS y no se pueden desplegar en los nodos de Arm.\
- Este complemento se puede instalar en nodos que utilizan containerd o el motor de contenedor de Docker. En los nodos de containerd, puede rastrear las actualizaciones de pod en tiempo real. En los nodos de Docker, puede consultar actualizaciones de pod en modo de sondeo.
- Solo se pueden recopilar estadísticas de tráfico de contenedores seguro (Kata como el tiempo de ejecución contenedor) y contenedores comunes (runC como el tiempo de ejecución contenedor) en un clúster de CCE Turbo.
- Después de instalar el complemento, el tráfico no se supervisa de forma predeterminada. Es necesario crear un CR para configurar una tarea de supervisión para la supervisión del tráfico.
- Asegúrese de que hay suficientes recursos en un nodo para instalar el complemento.
- El origen de las etiquetas de supervisión y de las etiquetas de usuario debe estar disponible antes de crear un pod.

### Instalación del complemento

**Paso 1** Inicie sesión en la consola de CCE. En el panel de navegación, elija **Add-ons**. En la página **Add-ons**, haga clic en **Install** bajo **dolphin**.

**Paso 2** En la página **Install Add-on**, seleccione un clúster en el paso **Basic Information**.

----Fin

### Entrega de una tarea de supervisión

Puede entregar una tarea de supervisión mediante la creación de un CR. Actualmente, se puede crear un CR invocando a una API o usando el comando **kubectl apply** después de

iniciar sesión en un nodo de trabajo. En versiones posteriores, se puede crear un CR en la consola. Un CR representa una tarea de monitorización y proporciona parámetros opcionales como **selector**, **podLable** y **ip4Tx**. Para obtener más información, consulte la plantilla de creación de CR a continuación.

```
apiVersion: crd.dolphin.io/v1
kind: MonitorPolicy
metadata:
  name: example-task           # Monitoring task name.
  namespace: kube-system      # The value must be kube-system. This field is
                              # mandatory.
spec:
  selector:                   # (Optional) Backend monitored by the dolphin
                              # add-on, for example, labelSelector. By default, all containers on the node are
                              # monitored.
  matchLabels:
    app: nginx
  matchExpressions:
    - key: app
      operator: In
      values:
        - nginx
  podLable: [app]             # (Optional) Pod label.
  ip4Tx:                      # (Optional) Indicates whether to collect
                              # statistics about the number of sent IPv4 packets and the number of sent IPv4
                              # bytes. This function is disabled by default.
    enable: true
  ip4Rx:                      # (Optional) Indicates whether to collect
                              # statistics about the number of received IPv4 packets and the number of received
                              # IPv4 bytes. This function is disabled by default.
    enable: true
  ip4TxInternet:             # (Optional) Indicates whether to collect
                              # statistics about the number of sent IPv4 packets and the number of sent IPv4
                              # bytes. This function is disabled by default.
    enable: true
```

**PodLable:** Puede introducir las etiquetas de varios pods y separarlos con comas (,), por ejemplo, [app, version].

Las etiquetas deben cumplir con las siguientes reglas. La expresión regular correspondiente es  $(^[a-zA-Z_]\$)|(^([a-zA-Z][a-zA-Z0-9_][a-zA-Z0-9])([a-zA-Z0-9_]){0,254}\$)$ .

- Se puede introducir un máximo de cinco etiquetas. Cada etiqueta contiene un máximo de 256 caracteres.
- El valor no puede comenzar con un dígito o guiones bajos dobles (\_).
- El formato de una sola etiqueta debe cumplir con A-Za-z\_0-9.

#### Ejemplo 1

```
apiVersion: crd.dolphin.io/v1
kind: MonitorPolicy
metadata:
  name: example-task
  namespace: kube-system
spec:
  podLable: [app]
  ip4Tx:
    enable: true
```

En el ejemplo anterior, el nombre de la tarea de supervisión es **example-task**, que supervisa todos los pods de un nodo y genera el número de paquetes de IPv4 enviados y el número de bytes enviados. Si el contenedor monitorizado contiene la etiqueta **app**, la información de clave-valor de la etiqueta correspondiente se lleva en las métricas de monitorización. De lo contrario, el valor de la etiqueta correspondiente es **not found**.

#### Ejemplo 2

```

apiVersion: crd.dolphin.io/v1
kind: MonitorPolicy
metadata:
  name: example-task
  namespace: kube-system
spec:
  selector:
    matchLabels:
      app: nginx
  podLabel: [test, app]
  ip4Tx:
    enable: true
  ip4Rx:
    enable: true
  ip4TxInternet:
    enable: true
    
```

En el ejemplo anterior, el nombre de la tarea de monitorización es **example-task**, que monitoriza todos los pods que cumplen con el labelector con `app=nginx` en un nodo y genera las seis métricas. Si el contenedor monitorizado contiene etiquetas **test** y **app**, la información clave-valor de la etiqueta correspondiente se lleva en las métricas de monitorización. De lo contrario, el valor de la etiqueta correspondiente es **not found**.

Puede crear, modificar y eliminar tareas de supervisión en el formato anterior. Actualmente, se pueden crear un máximo de 10 tareas de monitorización. Cuando varias tareas de monitoreo coinciden con el mismo backend de monitoreo, cada backend de monitoreo genera la métrica de monitoreo específica para el número de tareas de monitoreo.

 **NOTA**

- Si modifica o elimina una tarea de supervisión, se perderán los datos de supervisión recopilados por la tarea de supervisión. Por lo tanto, realice esta operación con precaución.
- Después de desinstalar el complemento, se quita el CR de la tarea de supervisión junto con el complemento.

## Comprobación de estadísticas de tráfico

Los datos de seguimiento recopilados por este complemento se exportan en formato de exportador de Prometheus, que se pueden obtener de cualquiera de las siguientes maneras:

- Instale el complemento de prometheus, que se interconecta automáticamente con el complemento dolphin y recopila periódicamente información de monitoreo.
- Acceda directamente al puerto de servicio 10001 proporcionado por el complemento dolphin, por ejemplo, `http://{POD_IP}:10001/metrics`.

Tenga en cuenta que si accede al puerto de servicio dolphin en un nodo, debe permitir el acceso desde el grupo de seguridad del nodo y el pod.

Puede instalar el complemento prometheus para ver la información de supervisión. Para obtener más información sobre cómo usar el complemento de prometheus, consulte [Monitoreo de métricas personalizadas con prometheus](#).

**Tabla 14-32** Métricas de monitorización admitidas

Métrica	Parámetro
Número de paquetes de IPv4 enviados a la red pública	ip4_send_pkt_internet

Métrica	Parámetro
Número de bytes de IPv4 enviados a la red pública	ip4_send_byte_internet
Número de paquetes de IPv4 recibidos	ip4_rcv_pkt
Número de bytes de IPv4 recibidos	ip4_rcv_byte
Número de paquetes de IPv4 enviados	ip4_send_pkt
Número de bytes de IPv4 enviados	ip4_send_byte

- Ejemplo 1 (número de paquetes de IPv4 enviados a la red pública):

```
dolphin_ip4_send_pkt_internet{app="nginx",pod="default/nginx-66c9c65dbf-zjg24",task="kube-system/example-task "} 241
```

En el ejemplo anterior, el espacio de nombres del pod es **default**, el nombre del pod es **nginx-66c9c65dbf-zjg24**, la etiqueta es **app** y el valor es **nginx**. Esta métrica se crea mediante **example-task** de tareas de supervisión, y el número de paquetes de IPv4 enviados por el pod a la red pública es de **241**.

- Ejemplo 2 (número de bytes de IPv4 enviados a la red pública):

```
dolphin_ip4_send_byte_internet{app="nginx",pod="default/nginx-66c9c65dbf-zjg24",task="kube-system/example-task "} 23618
```

En el ejemplo anterior, el espacio de nombres del pod es **default**, el nombre del pod es **nginx-66c9c65dbf-zjg24**, la etiqueta es **app** y el valor es **nginx**. Esta métrica se crea mediante **example-task** de tareas de supervisión, y el número de bytes de IPv4 enviados por el pod a la red pública es **23618**.

- Ejemplo 3 (número de paquetes de IPv4 enviados):

```
dolphin_ip4_send_pkt{app="nginx",pod="default/nginx-66c9c65dbf-zjg24",task="kube-system/example-task "} 379
```

En el ejemplo anterior, el espacio de nombres del pod es **default**, el nombre del pod es **nginx-66c9c65dbf-zjg24**, la etiqueta es **app** y el valor es **nginx**. Esta métrica se crea mediante las tareas de supervisión **example-task**, y el número de paquetes de IPv4 enviados por el pod es **379**.

- Ejemplo 4 (número de bytes de IPv4 enviados):

```
dolphin_ip4_send_byte{app="nginx",pod="default/nginx-66c9c65dbf-zjg24",task="kube-system/example-task "} 33129
```

En el ejemplo anterior, el espacio de nombres del pod es **default**, el nombre del pod es **nginx-66c9c65dbf-zjg24**, la etiqueta es **app** y el valor es **nginx**. Esta métrica se crea mediante las tareas de supervisión **example-task**, y el número de bytes de IPv4 enviados por el pod es **33129**.

- Ejemplo 5 (número de paquetes de IPv4 recibidos):

```
dolphin_ip4_rcv_pkt{app="nginx",pod="default/nginx-66c9c65dbf-zjg24",task="kube-system/example-task "} 464
```

En el ejemplo anterior, el espacio de nombres del pod es **default**, el nombre del pod es **nginx-66c9c65dbf-zjg24**, la etiqueta es **app** y el valor es **nginx**. Esta métrica se crea mediante las tareas de supervisión **example-task**, y el número de paquetes de IPv4 recibidos por el pod es **464**.

- Ejemplo 6 (número de bytes de IPv4 recibidos):

```
dolphin_ip4_rcv_byte{app="nginx",pod="default/nginx-66c9c65dbf-zjg24",task="kube-system/example-task "} 34654
```

En el ejemplo anterior, el espacio de nombres del pod es **default**, el nombre del pod es **nginx-66c9c65dbf-zjg24**, la etiqueta es **app** y el valor es **nginx**. Esta métrica se crea mediante las tareas de supervisión **example-task**, y el número de bytes de IPv4 recibidos por el pod es **34654**.

 **NOTA**

Si el contenedor no contiene la etiqueta especificada, el valor de la etiqueta en el cuerpo de la respuesta es **not found**. El formato es el siguiente:

```
dolphin_ip4_send_byte_internet{test="not found", pod="default/nginx-66c9c65dbf-zjg24",task="default" } 23618
```

## Historial de cambios

**Tabla 14-33** Versiones de complementos de CCE

Versión del complemento	Versión de clúster admitida
1.1.8	/v1.(19 21 23 25).*/
1.1.6	/v1.(19 21 23).*/
1.1.5	/v1.(19 21 23).*/
1.1.2	/v1.(19 21 23).*/
1.0.1	/v1.(19 21).*/

## 14.17 e-backup

### Presentación

e-backup admite respaldo y restauración de clústeres. Realiza copias de respaldo de los datos de las aplicaciones y los datos de servicio en OBS y proporciona copias de respaldo de datos locales y remotos.

### Restricciones

- No agregue, elimine ni modifique el clúster durante la copia de respaldo/restauración. De lo contrario, la copia de respaldo/restauración puede fallar o quedar incompleta.
- Si cambia el clúster, se recomienda esperar 15 minutos hasta que el clúster esté estable y, a continuación, realizar la operación de copia de respaldo.
- Cuando se utilizan instantáneas de disco de EVS para la copia de respaldo, solo se admiten los PV de EVS y se aplican las restricciones de instantáneas (por ejemplo, no se admite la restauración entre las AZ). Los precios son los mismos que los de las instantáneas de disco de EVS.
- Cuando se utiliza restic para la copia de respaldo, los datos de EVS, SFS, SFS Turbo y PV de OBS se copian y se cargan en el repositorio de copia de respaldo de OBS.
- restic crea una instantánea para los datos en el punto de tiempo de copia de respaldo y carga los datos, lo que no afecta a la lectura y escritura de datos posteriores. Sin

embargo, restic no verifica el contenido del archivo y la coherencia del servicio. Se aplican restricciones de restricción.

- La memoria ocupada por restic está relacionada con el tamaño de los datos de PV respaldados por primera vez. Si el tamaño de los datos es superior a 500 GB, se recomienda utilizar los métodos de migración proporcionados por los servicios de almacenamiento en la nube. Si utiliza este complemento, puede modificar las cuotas de recursos del contenedor estérico haciendo referencia a la guía de operaciones.
- Puede utilizar Hooks para garantizar la coherencia de los datos de servicio para las aplicaciones con estado durante la copia de respaldo, por ejemplo, sincronizar los datos de memoria con los archivos.
- Durante la restauración, puede ajustar las configuraciones para adaptarse a las diferencias del entorno antes y después de la migración.
  - Una aplicación se puede restaurar desde el espacio de nombres original a otro espacio de nombres especificado. Sin embargo, debe confirmar que no se accede a la aplicación con un Service fijo durante la restauración.
  - Puede cambiar la dirección de la imagen (repo) de la aplicación a otra ruta de la imagen. El nombre y la etiqueta de la imagen permanecen sin cambios durante la restauración.
  - Puede cambiar el nombre de la clase de almacenamiento utilizada por la aplicación a una nueva. Tenga en cuenta que los recursos de almacenamiento de backend deben ser del mismo tipo, por ejemplo, desde el almacenamiento de bloques hasta el almacenamiento de bloques.
- Se aplican restricciones de Velero y de restic. Por ejemplo, durante la restauración, el Service borrará ClusterIP para adaptarse mejor a las diferencias entre los clústeres de Kubernetes de origen y destino.

## Instalación del complemento

**Paso 1** Inicie sesión en la consola de CCE. En el panel de navegación, elija **Add-ons**. Localice el complemento de copia de respaldo electrónica y haga clic en **Install**.

**Paso 2** En la página **Install Add-on**, seleccione el clúster, defina los parámetros y haga clic en **Install**.

Se admite el siguiente parámetro:

**volumeWorkerNum**: número de trabajos de copia de respaldo de volúmenes simultáneos. El valor predeterminado es **3**.

----**Fin**

## Uso del complemento

e-backup utiliza los bucket de OBS como ubicación de almacenamiento de copia de respaldo. Antes de realizar una copia de respaldo de los datos, debe realizar operaciones de **Preparación de claves** y **Creación de una ubicación de almacenamiento**.

Las copias de respaldo pueden ser **inmediatas** y **programadas**. Las restauraciones pueden ser **inmediatas**.



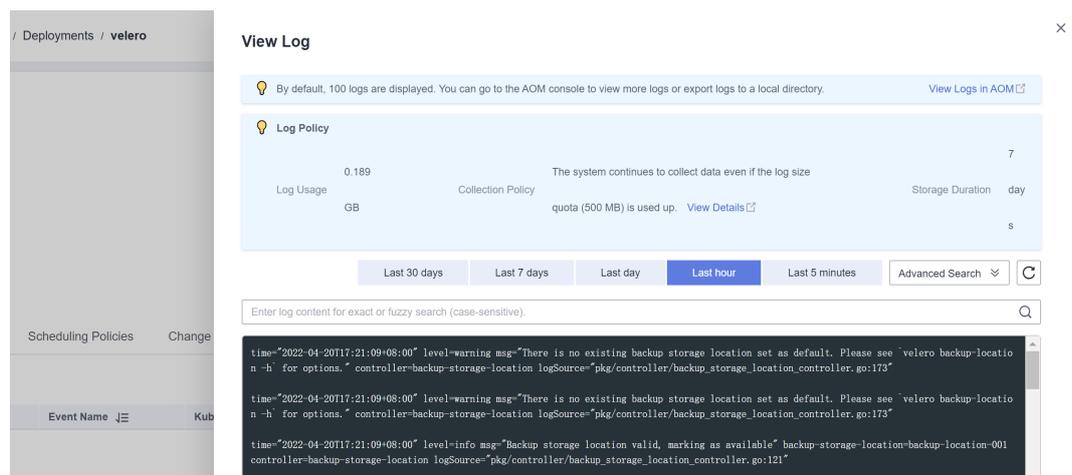
```
objectStorage:
  bucket: tools-cce      # OBS bucket name
  prefix: for-backup    # Subpath name
  provider: huawei       # Uses the OBS service.
```

- El campo **prefix** es opcional y otros campos son obligatorios. El valor de **provider** se fija en **huawei**.
- Puede obtener el punto de conexión de **Regiones y puntos de conexión**. Asegúrese de que todos los nodos del clúster puedan acceder al punto de conexión. Si el punto de conexión no lleva una cabecera de protocolo (http o https), se utiliza **https** por defecto.
- Establezca correctamente **name** y **key** en la credencial. De lo contrario, la copia de respaldo electrónica no puede acceder a la ubicación de almacenamiento.

Una vez completada la creación, espere 30 segundos para comprobar y sincronizar la ubicación de almacenamiento de copia de respaldo. A continuación, compruebe si **PHASE** es de tipo **Available**. La ubicación solo está disponible cuando el valor es **Available**.

```
$ kubectl get backupstoragelocations.velero.io backup-location-001 -n velero
NAME                PHASE      LAST VALIDATED  AGE    DEFAULT
backup-location-001 Available   23s             23m
```

Si **PHASE** no es **Available** durante mucho tiempo, puede ver los logs de copia de respaldo electrónica para localizar el fallo. Una vez instalada la copia de respaldo electrónica, se crea una carga de trabajo llamada **velero** en el espacio de nombres **velero** que se registra en los logs de velero.



## Copia de respaldo inmediata

El proceso de copia de respaldo se inicia inmediatamente y se detiene al finalizar. Este modo se utiliza comúnmente para la clonación y la migración.

Puede usar el manifiesto de copia de respaldo que aparece a continuación y ejecutar **kubectl create** para crear una tarea de copia de respaldo.

```
apiVersion: velero.io/v1
kind: Backup
metadata:
  name: backup-01
  namespace: velero
spec:
  includedNamespaces:
    - nginx
    - mysql
  labelSelector:
```

```
matchExpressions:
- key: direction
  operator: In
  values:
  - back
  - front
matchLabels:
  app: nginx
  backup: velero
runMode: Normal
appData:
  volumes: Restic
hooks:
  resources:
  - name: hook01
    includedNamespaces:
    - nginx
    labelSelector: {}
    pre:
    - exec:
      command:
      - /bin/sh
      - -c
      - echo hello > hello.txt && echo goodbye > goodbye.txt
      container: container-0
      onError: Fail
      timeout: 30s
    post:
    - exec:
      command:
      - /bin/sh
      - -c
      - echo hello > hello.txt && echo goodbye > goodbye.txt
      container: container-0
      onError: Fail
      timeout: 30s
storageLocation: backup-location-001
ttl: 720h0m0s
```

#### Descripción de parámetros:

- Parámetros de copia de respaldo
  - **storageLocation**: (**obligatorio**) nombre de la ubicación de almacenamiento de copia de respaldo donde se almacenan los datos que se van a hacer copia de respaldo.
  - **ttl**: duración para almacenar las copias de respaldo en la ubicación, después de la cual se eliminan las copias de respaldo. El valor debe estar en el formato especificado. **h**, **m** y **s** indican hora, minuto y segundo, respectivamente. Por ejemplo, **24h** indica un día, y **3h4m5s** indica tres horas, cuatro minutos y cinco segundos. El valor predeterminado es 30 días (**720h0m0s**).
- Filtrado de recursos: los siguientes parámetros se utilizan como filtros. La intersección de estos campos, si todos están configurados, se utiliza para filtrar todos los recursos del clúster.
  - **includedNamespaces** y **excludedNamespaces**: si se deben realizar copias de respaldo de los recursos en ciertos espacios de nombres. Estos dos parámetros entran en conflicto entre sí. Elija uno para configurar. De forma predeterminada, todos los espacios de nombres están seleccionados.
  - **labelSelector**: hace copias de respaldo de los recursos con etiquetas específicas. El principio de funcionamiento es el mismo que en Kubernetes.
  - **runMode**: (**obligatorio**) modo de copia de respaldo. Las opciones de valor incluyen **Normal** (copia de respaldo de aplicaciones y datos), **AppOnly** (solo copia

de respaldo de aplicaciones), **DataOnly** (solo copia de respaldo de datos) y **DryRun** (no copia de respaldo de aplicaciones y datos; solo para verificación).

- Respaldo de datos de Service: los datos de servicio generados se pueden respaldar con copias instantáneas siempre (soportado solo cuando los EVS PVs como los volúmenes de datos) y backups restic (que respaldan todos los volúmenes de datos excepto los hostPath). Estos dos modos se pueden usar juntos.
  - **appData**: Modo de copia de seguridad de datos de PV. El valor puede ser **Restic** o **Snapshot** (no se usa por defecto). El modo **Snapshot** solo tiene efecto cuando el almacenamiento admite instantáneas y el complemento de instantáneas de CSI se despliega en el clúster.
- **hook**: Los ganchos son los comandos ejecutados antes o después de una copia de respaldo para gestionar con precisión sus copias de respaldo. Un hook es similar al comando **kubectrl exec** y solo se aplica a pods.
  - **includedNamespaces** y **excludedNamespaces**: si se debe ejecutar un gancho en los pods en ciertos espacios de nombres. Estos dos parámetros entran en conflicto entre sí. Elija uno para configurar. De forma predeterminada, todos los espacios de nombres están seleccionados.
  - **labelSelector**: ejecuta un gancho en los pods con ciertas etiquetas. El principio de funcionamiento es el mismo que en Kubernetes.
  - **command**: comando a ejecutar.
  - **contenedor**: nombre del contenedor en el que se ejecuta el comando. El valor predeterminado es el primer contenedor cuando hay varios contenedores en el pod.
  - **onError**: acción a realizar cuando el gancho no se ejecuta. El valor puede ser **Continue** o **Fail**. El valor predeterminado es **Fail**.
  - **Continue** indica que las operaciones subsiguientes continúan independientemente de los fallos de ejecución de gancho. **Fail** indica que las operaciones posteriores no continuarán cuando se produzca un error de ejecución de gancho.
  - **timeout**: tiempo de espera de ejecución del gancho, después del cual el gancho falla. El valor predeterminado es 30s.

Las fallas del gancho afectan solo a los pods. La copia de respaldo de otros objetos como los Services no se ve afectada.

Los ganchos no están disponibles en todo el mundo. Si el pod para ejecutar un gancho no está seleccionado como el objeto de copia de respaldo, el gancho no se ejecutará. Se puede considerar que se filtran más los objetos a los que se va a realizar una copia de respaldo con **includedNamespaces** o **excludedNamespaces**.

 **NOTA**

Todos los elementos configurables se describen anteriormente. A continuación se proporcionan algunas **sugerencias de configuración de copia de respaldo**.

- Conservar las copias de respaldo por día (24 horas).
- Utilizar **includeNamespace** para especificar el ámbito de copia de respaldo, ya que en la mayoría de los casos, las aplicaciones se despliegan en un espacio de nombres específico. Utilizar **labelSelector** si necesita controlar los objetos de copia de respaldo con mayor precisión. Antes de esto, todos los objetos de destino deben tener etiquetas correspondientes. Usar **includeNamespace** y **labelSelector** juntos puede satisfacer la mayoría de los escenarios.
- Al usar Restic para hacer una copia de respaldo de los datos del servicio, si no está familiarizado con el modo OUT/IN, puede omitir agregar anotaciones a los pods que requieren una copia de respaldo de volumen. En su lugar, establezca **defaultVolumesToRestic** en **true** para hacer una copia de respaldo de los datos de servicio de los volúmenes de pod. El valor **false** indica que no hay copias de respaldo.
- Utilizar ganchos para controlar con precisión sus copias de respaldo. Evitar las tareas que se ejecutan durante mucho tiempo. No operar directamente el sistema de archivos cuando ejecutar los comandos en el gancho.

Una vez completada la copia de respaldo, ejecute los siguientes comandos para ver el estado de la copia de respaldo (**status**):

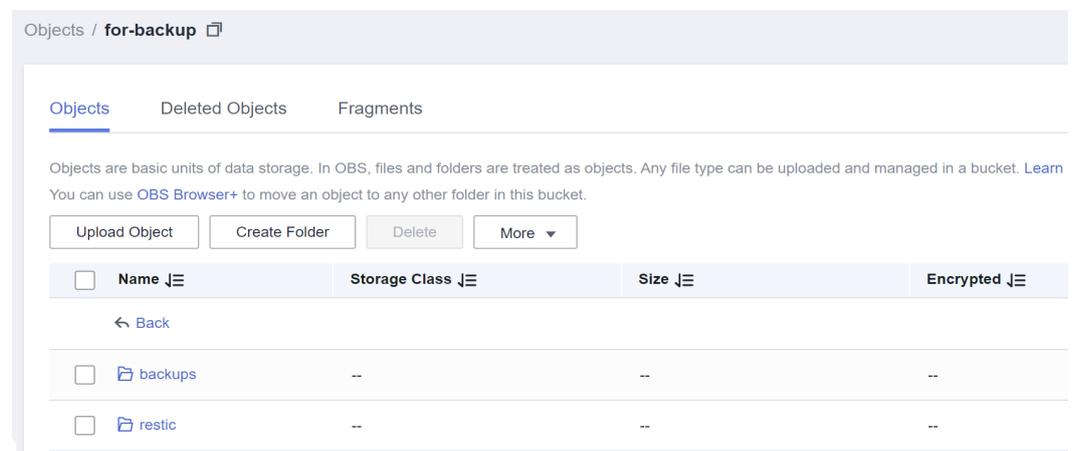
```
$ kubectl -n velero get backups backup-01 -o yaml | grep "phase"
  phase: Completed

$ kubectl -n velero get backups backup-01 -o yaml
.....
status:
  .....
```

Estado de copia de respaldo

- **FailedValidation**: El manifiesto de copia de respaldo está configurado incorrectamente. Compruebe **Backup.Status.ValidationErrors** para encontrar la causa.
- **InProgress**: La copia de respaldo está en curso.
- **Completed**: La copia de respaldo está completa y no se produce ningún error.
- **PartiallyFailed**: La copia de respaldo está completa, pero se produce un error (como un error de ejecución de gancho) durante la copia de respaldo de ciertos objetos.
- **Failed**: La copia de respaldo falla y se produce un error que afecta a todo el proceso.
- **Deleting**: La copia de respaldo se está eliminando.

Una vez completada la copia de respaldo inicial, las carpetas **backups** y **restic** se muestran en el bucket de OBS.



**Los registros de copia de respaldo se almacenan en un bucket de OBS.** Suponga que el nombre de la copia de respaldo es **backup-001**. Vaya a la consola de OBS, busque la ubicación de almacenamiento basada en el nombre del bucket configurado y el nombre de la ruta secundaria, vaya al directorio **backups/backup-01** y busque el archivo **backup-01-logs.gz**. A continuación, descargue, descomprima y vea los registros.

## Copia de seguridad periódica

Se realiza una copia de respaldo de los datos periódicamente según lo configurado. Este modo se utiliza comúnmente para la recuperación ante desastres.

Puede usar el manifiesto de planificación que aparece a continuación y ejecutar el comando **kubectrl create** para crear una planificación. Puede etiquetar la programación según sea necesario. Las etiquetas que agregue en el manifiesto se adjuntarán a las copias de respaldo creadas por la programación. Después de crear una programación en un clúster, se realiza una copia de respaldo inmediatamente. A continuación, se realiza una copia de respaldo de los datos periódicamente según se especifica.

```
apiVersion: velero.io/v1
kind: Schedule
metadata:
  name: schedule-backup-001
  namespace: velero
spec:
  schedule: 0 */10 * * *
  template:
    runMode: Normal
    hooks: {}
    includedNamespaces:
      - nginx
      - mysql
    labelSelector:
      matchExpressions:
        - key: direction
          operator: In
          values:
            - back
            - front
      matchLabels:
        app: nginx
        backup: velero
    storageLocation: backup-location-001
    ttl: 720h0m0s
```

Descripción de parámetros:

- **schedule**: tiempo de ejecución de copias de respaldo periódicas. El formato **@every** y las expresiones cron estándar de Linux son compatibles.
  - **@every NUnit**: **N** es un entero positivo. Las unidades **s**, **m** o **h** permanecen en reposo durante segundos, minutos y horas, respectivamente. Por ejemplo, **@every 2h30m** indica que la copia de respaldo se activa cada 2 horas y 30 minutos.
  - Expresión de Cron: Los cinco valores representan minutos, horas, día del mes, mes y día de la semana, respectivamente.
- **template**: manifiesto de copia de respaldo, que es el mismo que **spec** de [Copia de respaldo inmediata](#).

## Eliminación de una copia de seguridad

Puede eliminar los objetos de copia de respaldo y los objetos relacionados (como copias de respaldo, restauraciones y programaciones) de un clúster y eliminar las copias de respaldo de

la ubicación de almacenamiento cuando se genera una gran cantidad de datos de copia de respaldo.

Puede usar el manifiesto `DeleteBackupRequest` a continuación y ejecutar el comando `kubectl create` para crear una solicitud de eliminación de copia de respaldo.

```
apiVersion: velero.io/v1
kind: DeleteBackupRequest
metadata:
  name: backup-001-delete
  namespace: velero
spec:
  backupName: backup-001 # Name of the backup to be deleted.
```

Consulte el estado.

```
$ kubectl -n velero get deletebackuprequests backup-001-delete -o yaml | grep "
phase"
  phase: InProgress
```

- **InProgress:** La tarea de eliminación está en curso.
- **Processed:** Se ha procesado la tarea de eliminación.

### ATENCIÓN

- El estado **Processed** indica que la copia de respaldo electrónica ha procesado la tarea pero puede que no la complete. Puede comprobar los errores en el campo `deletebackuprequest.status.errors`. Si e-backup procesa correcta y completamente la tarea de eliminación, también se elimina el objeto **DeleteBackupRequest**.
- No elimine manualmente el contenido en la ubicación de almacenamiento (bucket de OBS).

## Restauración inmediata

Utilice una copia de respaldo inmediata como origen de datos y restaure los datos en otro espacio de nombres o clúster. Este modo se aplica a todos los escenarios.

Puede utilizar el manifiesto `Restore` a continuación y ejecutar el comando `kubectl create` para crear una solicitud de eliminación de copia de respaldo.

```
apiVersion: velero.io/v1
kind: Restore
metadata:
  name: restore-01
  namespace: velero
spec:
  backupName: backup-01
  hooks:
    resources:
      - name: restore-hook-1
        includedNamespaces:
          - mysql
        labelSelector: {}
        postHooks:
          - init:
              initContainers:
                - name: restore-hook-init1
                  image: alpine:latest
                  volumeMounts:
                    - mountPath: /restores/pvc1-vm
                      name: pvc1-vm
```

```

        command:
        - /bin/ash
        - -c
        - echo -n "FOOBARBAZ" >> /restores/pvc1-vm/foobarbaz
    - name: restore-hook-init2
      image: alpine:latest
      volumeMounts:
        - mountPath: /restores/pvc2-vm
          name: pvc2-vm
      command:
        - /bin/ash
        - -c
        - echo -n "DEADFEED" >> /restores/pvc2-vm/deadfeed
    - exec:
      execTimeout: 1m
      waitTimeout: 5m
      onError: Fail
      container: mysql
      command:
        - /bin/bash
        - '-c'
        - 'while ! mysql_isready; do sleep 1; done'
    - exec:
      container: mysql
      waitTimeout: 6m
      execTimeout: 1m
      onError: Continue
      command:
        - /bin/bash
        - '-c'
        - 'mysql < /backup/backup.sql'
  includedNamespaces:
  - nginx
  - mysql
  namespaceMapping:
    nginx: nginx-another
    mysql: mysql-another
  labelSelector: {}
  preserveNodePorts: false
  storageClassMapping:
    disk: csi-disk
    obs: csi-obs
  imageRepositoryMapping:
    quay.io/coreos: swr.ap-southeast-1.myhuaweicloud.com/everest

```

Descripción de parámetros:

- Fuente de datos
  - backupName:** (obligatorio) copia de respaldo inmediata que se utiliza como origen de datos.
- Parámetros de filtrado de recursos: similares a los de [Copia de respaldo inmediata](#).
- Procesamiento personalizado
  - **namespaceMapping:** restaura los datos de copia de respaldo en otro espacio de nombres. El valor es una asignación en el formato de *Source: Target*. No es necesario que el nuevo espacio de nombres exista en el clúster de destino.
  - **storageClassMapping:** cambia el storageClassName usado por los recursos de copia de respaldo como PV y PVC. Los tipos storageClass deben ser los mismos.
  - **imageRepositoryMapping:** cambia el campo **images** de la copia de respaldo. Se utiliza para la asignación de repositorios, excluidos el cambio del nombre de la imagen y la etiqueta (para evitar que la migración y la actualización se acoplen). Por ejemplo, después de migrar **quay.io/coreos/etcd:2.5** a SWR, puede usar **swr.ap-southeast-1.myhuaweicloud.com/everest/etcd:2.5** en el repositorio de

imágenes local. El formato de configuración es el siguiente: **quay.io/coreos:swr.ap-southeast-1.myhuaweicloud.com/everest**

- **preserveNodePorts**: Si establece este parámetro en **false**, el sistema conservará únicamente los nodePorts configurados, no los generados automáticamente por el Service.
- **hooks**: Puede agregar ganchos de inicio (usados para agregar initContainers al pod) y ganchos exec (usados para ejecutar algunos comandos). Para obtener más información sobre cómo configurar un gancho de inicio, consulte la definición de initContainers en Kubernetes. A continuación se describe la configuración general del gancho y los parámetros de un gancho exec.
  - **includedNamespaces** y **excludedNamespaces**: si se debe ejecutar un gancho en los pods en ciertos espacios de nombres. Estos dos parámetros entran en conflicto entre sí. Elija uno para configurar. De forma predeterminada, todos los espacios de nombres están seleccionados.
  - **labelSelector**: ejecuta un gancho en los pods con ciertas etiquetas. El principio de funcionamiento es el mismo que en Kubernetes.
  - **command**: comando a ejecutar.
  - **contenedor**: nombre del contenedor en el que se ejecuta el comando. El valor predeterminado es el primer contenedor cuando hay varios contenedores en el pod.
  - **onError**: acción a realizar cuando el gancho no se ejecuta. El valor puede ser **Continue** o **Fail**. El valor predeterminado es **Fail**.
  - **Continue** indica que las operaciones subsiguientes continúan independientemente de los fallos de ejecución de gancho. **Fail** indica que las operaciones posteriores no continuarán cuando se produzca un error de ejecución de gancho.
  - **execTimeout**: tiempo de espera de ejecución del gancho, después del cual el gancho falla. El valor predeterminado es 30s.
  - **waitTimeout**: período de tiempo de espera desde el momento en que e-backup se prepara para ejecutar el gancho hasta el momento en que el contenedor comienza a ejecutar el gancho. Si se excede este período, el gancho falla. El valor predeterminado es 0s, lo que indica que no hay límite de tiempo de espera.

 **NOTA**

- Seleccione un origen de datos correcto y asegúrese de que la copia de respaldo está en el estado **Completed**.
- Establezca los parámetros relacionados con el filtrado de recursos solo cuando sea necesario.
- Los datos de Service se restauran mediante una copia de respaldo electrónica basada en el modo de copia de respaldo seleccionado. No se requieren las configuraciones u operaciones manuales.
- Para obtener más información sobre cómo usar ganchos, consulte las sugerencias de uso de *Immediate Backup*. Puede omitir **waitTimeout** a menos que sea necesario.
- Se recomienda restaurar lo que se ha respaldado en un nuevo espacio de nombres para evitar configuraciones erróneas que pueden deshabilitar la aplicación restaurada.

Una vez completada la restauración, ejecute los siguientes comandos para ver el estado de restauración (**status**):

```
$ kubectl -n velero get restores restore-01 -o yaml | grep " phase"
  phase: Completed

$ kubectl -n velero get restores restore-01 -o yaml
.....
status:
  .....
```

#### Descripción de estado

- **FailedValidation:** El manifiesto de restauración está configurado incorrectamente. Compruebe **Restore.Status.ValidationErrors** para encontrar la causa.
- **InProgress:** La restauración está en curso.
- **Completed:** La restauración está completa y no se produce ningún error.
- **PartiallyFailed:** La restauración está completa, pero se produce un error (como un error de ejecución de gancho) durante la restauración de ciertos objetos.
- **Failed:** Se produce un error de restauración y se produce un error que afecta a todo el proceso.

Compruebe los registros, avisos y errores generados durante la restauración.

Suponga que el nombre de restauración es **restore-01**. Vaya a la consola de OBS, busque la ubicación de almacenamiento basada en el nombre del bucket configurado y el nombre de la subruta, y vaya al directorio **restores/restore-01**. Existen los dos archivos siguientes:

- **restore-01-logs.gz:** archivo de log, que se puede descargar, descomprimir y ver.
- **restore-01-results.gz:** archivo de resultado de restauración, incluidos los avisos y errores.

## Historial de cambios

**Tabla 14-34** Versiones de complementos de CCE

Versión del complemento	Versión de clúster admitida
1.2.0	/v1.(15 17 19 21).*/

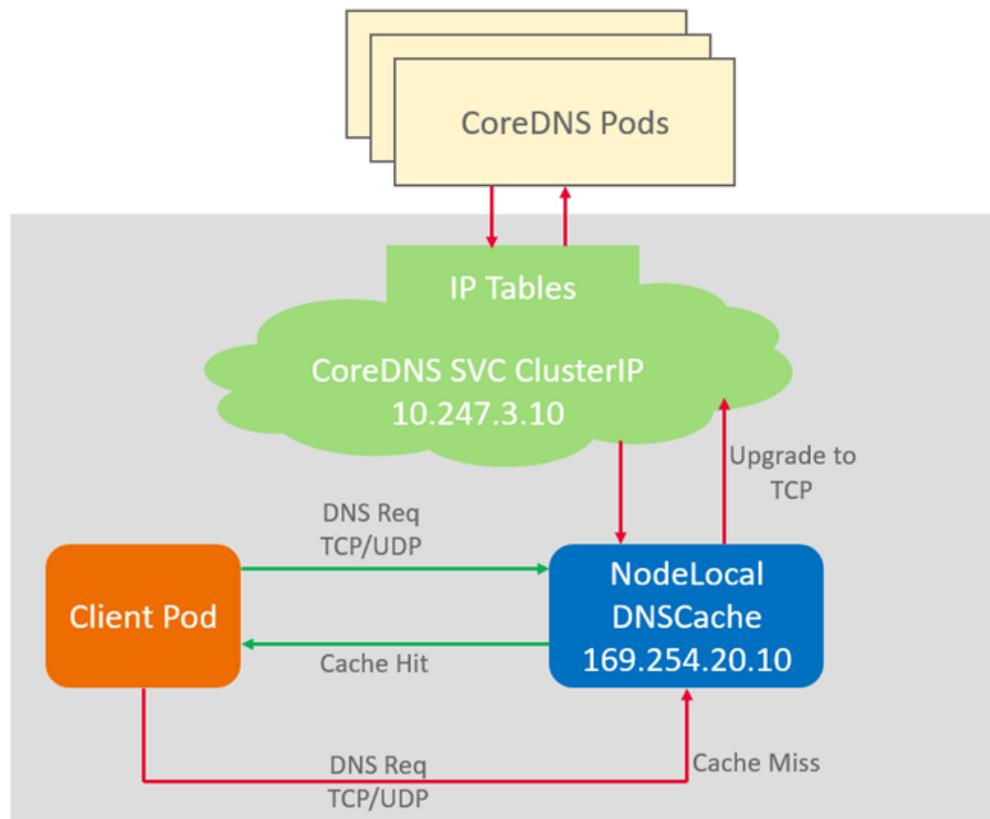
## 14.18 node-local-dns

### Presentación

NodeLocal DNSCache mejora el rendimiento de DNS del clúster al ejecutar proxys de caché de DNS como DaemonSets en los nodos del clúster.

Comunidad de código abierto: <https://github.com/kubernetes/dns>

Figura 14-11 Ruta de consulta de NodeLocal DNSCache



## Notas y restricciones

- Esta función solo está disponible para clústeres de v1.19 y versiones posteriores.

## Instalación del complemento

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster. Elija **Add-ons** en el panel de navegación, localice **node-local-dns** a la derecha y haga clic en **Install**.

**Paso 2** En la página **Install Add-on**, seleccione las especificaciones del complemento y establezca los parámetros relacionados.

### Especificaciones

- **Pods:** Establezca el número de pods en función de los requisitos de servicio.
- **Containers:** Establezca una cuota de contenedor adecuada en función de los requisitos de servicio.

### Parámetros

- **DNS Config Injection:** Al habilitar la inyección de configuración de DNS, se inyectará la dirección de nodo-local-dns en nuevos pods, sin necesidad de operación manual. Una vez activada esta función, se crea un contenedor denominado **node-local-dns-admission-controller** para activar la inyección automática. De lo contrario, el contenedor no se crea.

**Paso 3** Haga clic en **Install**.

----Fin

## Enlaces útiles

[Uso de DNSCache de NodeLocal para mejorar el rendimiento de DNS](#)

## Historial de cambios

Tabla 14-35 Versiones de complementos de CCE

Versión del complemento	Versión de clúster admitida	Versión de la comunidad (solo para clústeres de v1.17 y posteriores)
1.2.4	/v1.(19 21 23 25).*/	1.21.1
1.2.2	/v1.(19 21 23).*/	1.21.1

# 14.19 kube-prometheus-stack

## Presentación

kube-prometheus-stack utiliza Prometheus-operator y Prometheus para proporcionar capacidades de monitoreo de clúster de Kubernetes de extremo a extremo fáciles de usar.

Con este complemento, puede conectarse a Container Intelligent Analysis (CIA) para ver los datos de monitoreo y configurar alarmas.

Comunidad de código abierto: <https://github.com/prometheus/prometheus>

## Restricciones

De forma predeterminada, el componente kube-state-metrics del complemento no recopila etiquetas ni anotaciones de recursos de Kubernetes. Para recopilar estas etiquetas y anotaciones, debe habilitar manualmente la función de recopilación en los parámetros de inicio y comprobar si las métricas correspondientes se agregan a la lista blanca de recopilación de ServiceMonitor denominada **kube-state-metrics**. Para obtener más información, véase [Recopilación de todas las etiquetas y anotaciones de un pod](#).

## Instalación del complemento

- Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster. Elija **Add-ons** en el panel de navegación, localice **kube-prometheus-stack** a la derecha y haga clic en **Install**.
- Paso 2** En la página **Install Add-on**, configure las especificaciones.
  - **Containers:** instancia de componente creada por el complemento. Para obtener más información, véase [Componentes](#). Puede seleccionar o personalizar una especificación según sea necesario.
- Paso 3** Configuración de los parámetros relacionados.
  - **Connect to Third Party:** Para informar los datos de Prometheus a un sistema de monitoreo de terceros, debe introducir la dirección y el token del sistema de monitoreo de terceros y determinar si se omite la autenticación de certificado.

- **Prometheus HA:** Los componentes de Prometheus-server, Prometheus-operator, thanos-query, custom-metrics-apiserver y alertmanager se despliegan en modo de varias instancias en el clúster.
- **Install Grafana:** Use Grafana para visualizar los datos de monitoreo. Grafana crea un volumen de almacenamiento de 5 GiB por defecto. Desinstalar el complemento **no eliminará este volumen**. El nombre de usuario y la contraseña predeterminados para el primer inicio de sesión es **admin**. Se le pedirá que cambie la contraseña inmediatamente después de iniciar sesión.
- **Collection Period:** período de recogida de datos de seguimiento.
- **Data Retention:** período de conservación de los datos de seguimiento.
- **Storage:** seleccione el tipo y el tamaño del disco para almacenar los datos de supervisión. La desinstalación del complemento no eliminará este volumen.

 **NOTA**

Existe un PVC disponible llamado **pvc-prometheus-server** en el espacio de nombres **monitoring** y se utilizará como fuente de almacenamiento.

**Paso 4** Haga clic en **Install**.

----Fin

## Adición de reglas de métrica personalizadas

El complemento kube-prometheus-stack de la nueva versión no proporciona métricas personalizadas. Es decir, las reglas de recopilación de métricas ya no se configuran en el ConfigMap de configuración de adaptador de usuario (adapter-config en versiones anteriores). Necesita agregar reglas de recopilación de métricas. Para obtener más información sobre cómo agregar reglas, consulte [Descubrimiento de métricas y configuración de presentación](#). Si actualiza el complemento de una versión anterior a la nueva, las configuraciones originales se heredan y utilizan.

### AVISO

Para usar prometheus para monitorear las métricas personalizadas, la aplicación debe proporcionar una API de monitorización de métricas. Para obtener más información, véase [Recopilación de datos de monitorización de Prometheus](#).

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster. En el panel de navegación, elija **ConfigMaps and Secrets**.

**Paso 2** Cambie al espacio de nombres **monitoring**, busque el ConfigMap **user-adapter-config** (**adapter-config** en las versiones anteriores) en la página de fichas **ConfigMaps** y haga clic en **Update**.



**Paso 3** En **Data**, haga clic en **Edit** para el archivo **config.yaml** para agregar una regla de recopilación de métricas personalizada bajo el campo **rules**. Haga clic en **OK**.

Puede agregar varias reglas de recopilación agregando varias configuraciones en el campo **rules**. Para obtener más información, consulte [Descubrimiento de métricas y configuración de presentación](#).

Ejemplo de regla de métrica personalizada:

```
rules:
# The rule matches the accumulated cAdvisor metric in seconds.
- seriesQuery: '{__name__=~"^container_.*",container!="POD",namespace!="",pod!=""}'
  resources:
    # Specify pod and namespace resources.
    overrides:
      namespace:
        resource: namespace
      pod:
        resource: pod
  name:
    # Delete the container_ prefix and _seconds_total suffix, and use the content
    captured in .* as the metric name.
    matches: "^container_(.*)seconds_total$"
    # Query metrics. .Series and .LabelMatchers are available in the Go language.
    Use separators << and >> to avoid conflicts with the Prometheus query language.
    metricsQuery: 'sum(rate(<<.Series>>{<<.LabelMatchers>>,container!="POD"}) [2m])
    by (<<.GroupBy>>)'
```

**NOTA**

En el ejemplo anterior, solo se recopilan métricas básicas de pod. Para recopilar métricas personalizadas, consulta la [guía oficial](#) para agregar o modificar reglas.

**Update ConfigMap**

✕

Name:

Namespace:

Description:  0/255

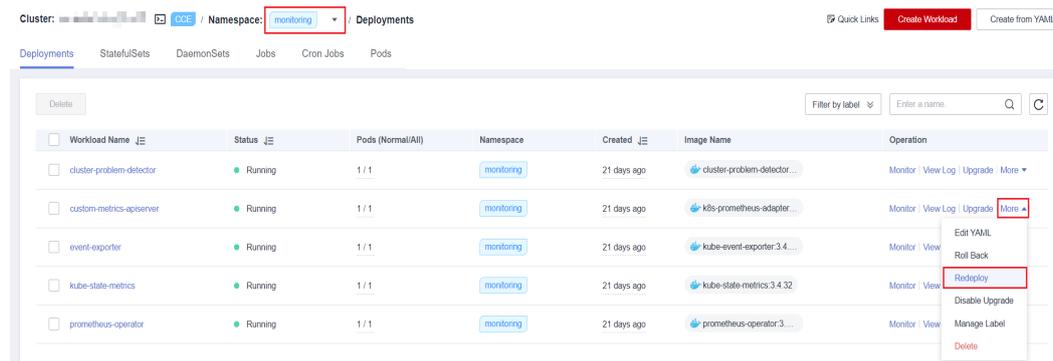
Data	Key	Value	Operation
	config.yaml	rules: - seriesQuery: '{__name__=~"^container_.*"...	<span style="border: 1px solid red; padding: 2px;">Edit</span> <span style="padding: 2px;">Delete</span>
+			

Label:  =

release = cceaddon-cie-collector

Version Management  Enabling this function allows ConfigMap data to be backed up before modification, so that you can easily manage and roll back data.

**Paso 4** Redistribuya la carga de trabajo **custom-metrics-apiserver** en el espacio de nombres **monitoring**.



---Fin

## Componentes

Todos los recursos de Kubernetes creados durante la instalación del complemento kube-prometheus-stack se crean en el espacio de nombres **monitoring**.

**Tabla 14-36** componentes kube-prometheus-stack

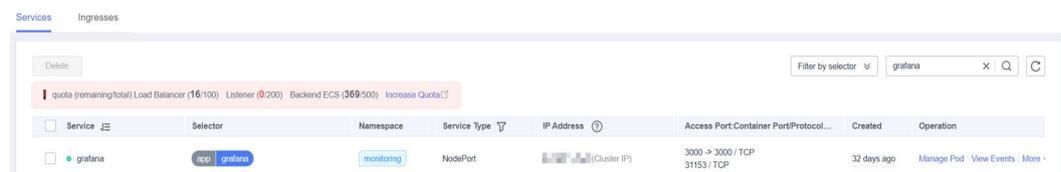
Componente del contenedor	Descripción	Tipo de recurso
prometheusOperator (nombre de la carga de trabajo: prometheus-operator)	Despliega y gestiona el Prometheus Server basado en CRD (CustomResourceDefinitions), y monitorea y procesa los eventos relacionados con estos CRD. Es el centro de control de todo el sistema.	Deployment
prometheus (nombre de la carga de trabajo: prometheus-server)	Un clúster de Prometheus Server desplegado por el operador basado en los CRD de Prometheus que puede considerarse StatefulSets.	StatefulSet
alertmanager (nombre de la carga de trabajo: alertmanager-alertmanager)	Centro de alarma del complemento. Recibe alarmas enviadas por Prometheus y gestiona la información de alarmas deduplicando, agrupando y distribuyendo.	StatefulSet
thanosSidecar	Disponible solo en modo HA. Se ejecuta con prometheus-server en el mismo pod para implementar el almacenamiento persistente de datos métricos de Prometheus.	Container
thanosQuery	Disponible solo en modo HA. Entrada para la consulta de PromQL cuando Prometheus está en escenarios de HA. Puede eliminar métricas duplicadas de Store o Prometheus.	Deployment

Componente del contenedor	Descripción	Tipo de recurso
adapter (nombre de carga de trabajo: custom-metrics-apiserver)	Agrega métricas personalizadas al servidor nativo de la API de Kubernetes.	Deployment
kubeStateMetrics (nombre de la carga de trabajo: kube-state-metrics)	Convierte los datos de métricas de Prometheus en un formato que las API de Kubernetes pueden identificar. De forma predeterminada, el componente kube-state-metrics no recopila todas las etiquetas y anotaciones de los recursos de Kubernetes. Para recopilar todas las etiquetas y anotaciones, consulte <a href="#">Recopilación de todas las etiquetas y anotaciones de un pod</a> .	Deployment
nodeExporter (nombre de la carga de trabajo: node-exporter)	Se despliega en cada nodo para recopilar datos de monitoreo de nodo.	DaemonSet
grafana (nombre de la carga de trabajo: grafana)	Visualiza los datos de monitoreo. grafana crea un volumen de almacenamiento de 5 GiB por defecto. La desinstalación del complemento no eliminará este volumen.	Deployment
clusterProblemDetector (nombre de la carga de trabajo: cluster-problem-detector)	Supervisa las excepciones del clúster.	Deployment

## Acceso a Grafana

Si Grafana se instala durante la instalación del complemento, puede acceder al nodo a través del Service llamado **grafana** que es un Service de NodePort. Si se accede al nodo desde una red externa, puede vincular una EIP al nodo y acceder al nodo a través del puerto del nodo.

Como se muestra en la siguiente figura, la dirección de acceso es **http://{{Node IP address}}:30913**.

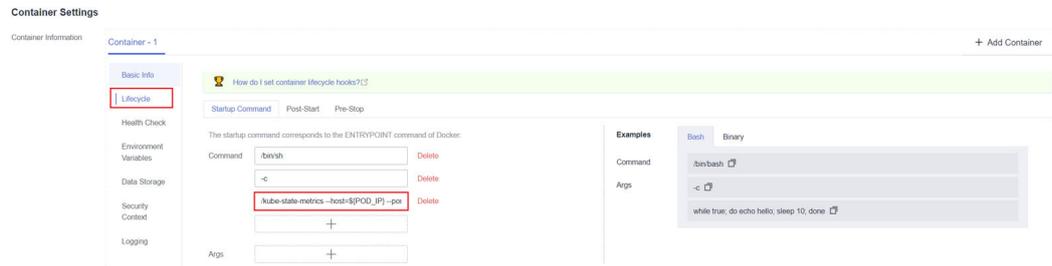


## Recopilación de todas las etiquetas y anotaciones de un pod

**Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster. En el panel de navegación, elija **Workloads**.

**Paso 2** Cambie al espacio de nombres **monitoring**, busque la carga de trabajo **kube-state-metrics** en la página de ficha **Deployments** y haga clic en **Upgrade** en la columna **Operation**.

**Paso 3** En el área **Lifecycle** de la configuración del contenedor, edite el comando de inicio.



Agregue la siguiente información al final del parámetro original de inicio **kube-state-metrics**:  
`--metric-labels-allowlist=pods=[*],nodes=[node,failure-domain.beta.kubernetes.io/zone,topology.kubernetes.io/zone]`

Para recopilar anotaciones, agregue parámetros en los parámetros de inicio de la misma manera.

`--metric-annotations-allowlist=pods=[*],nodes=[node,failure-domain.beta.kubernetes.io/zone,topology.kubernetes.io/zone]`

### AVISO

Al editar el comando de inicio, no modifique otros parámetros de inicio originales. De lo contrario, el componente puede ser anormal.

**Paso 4** **kube-state-metrics** comienza a recopilar las etiquetas/anotaciones de pods y nodos y comprueba si **kube\_pod\_labels/kube\_pod\_annotations** está en la tarea de recopilación de CloudScope.

```
kubectl get servicemonitor kube-state-metrics -nmonitoring -oyaml |
kube_pod_labels
```

----Fin

Para obtener más parámetros de inicio de kube-state-metrics, consulte [kube-state-metrics/cli-arguments](#).

## Proporcionar métricas de recursos

Las métricas de recursos de contenedores y nodos, como el uso de CPU y memoria, se pueden obtener con la API de métricas de Kubernetes. Se puede acceder directamente a las métricas de recursos, por ejemplo, mediante el comando **kubectl top**, o utilizar las políticas HPA o CustomedHPA personalizadas para el ajuste automático.

El complemento puede proporcionar la API de Kubernetes Metrics que está deshabilitada de forma predeterminada. Para habilitar la API, cree el siguiente objeto de APIService:

```
apiVersion: apiregistration.k8s.io/v1
kind: APIService
metadata:
  labels:
    app: custom-metrics-apiserver
    release: cceaddon-prometheus
    name: v1beta1.metrics.k8s.io
spec:
```

```
group: metrics.k8s.io
groupPriorityMinimum: 100
insecureSkipTLSVerify: true
service:
  name: custom-metrics-apiserver
  namespace: monitoring
  port: 443
version: v1beta1
versionPriority: 100
```

Puede guardar el objeto como un archivo, nombrarlo **metrics-apiservice.yaml** y ejecutar el siguiente comando:

```
kubectl create -f metrics-apiservice.yaml
```

Ejecute el comando **kubectl top**. Si se muestra la siguiente información, se puede acceder a la API de métricas:

```
# kubectl top pod -n monitoring
NAME                                                    CPU(cores)   MEMORY(bytes)
.....
custom-metrics-apiserver-d4f556ff9-12j2m             38m          44Mi
.....
```

### AVISO

Para desinstalar el complemento, ejecute el siguiente comando **kubectl** y elimine el objeto **APIService**. De lo contrario, el complemento de **metrics-server** no se puede instalar debido a los recursos **APIService** residuales.

```
kubectl delete APIService v1beta1.metrics.k8s.io
```

## Historial de cambios

**Tabla 14-37** Versiones de complementos de CCE

Versión del complemento	Versión de clúster admitida	Versión de la comunidad (solo para clústeres de v1.17 y posteriores)
3.5.1	/v1.(19 21 23).*/	<b>2.35.0</b>
3.5.0	/v1.(19 21 23).*/	<b>2.35.0</b>

## 14.20 storage-driver (complemento de recursos del sistema, descartado)

### Presentación

El controlador de almacenamiento funciona como un complemento estándar de FlexVolume de Kubernetes para permitir a contenedores usar recursos de almacenamiento de EVS, SFS, OBS y SFS Turbo. Al instalar y actualizar el controlador de almacenamiento, puede instalar y actualizar rápidamente las capacidades de almacenamiento en la nube.

**storage-driver es un complemento de recursos del sistema. Se instala de forma predeterminada cuando se crea un clúster de Kubernetes v1.13 o anterior.**

## Notas y restricciones

- Para clústeres creados en CCE, Kubernetes v1.15.11 es una versión transitoria en la que el complemento FlexVolume (storage-driver) es compatible con el complemento CSI (**everest**). Los clústeres de v1.17 y versiones posteriores ya no son compatibles con FlexVolume. Necesita usar el complemento más antiguo.
- El complemento FlexVolume será mantenido por los desarrolladores de Kubernetes, pero las nuevas funcionalidades solo se agregarán a CSI. Se aconseja no crear más almacenamiento que se conecte al complemento FlexVolume (storage-driver) en CCE. De lo contrario, los recursos de almacenamiento pueden no funcionar normalmente.
- Este complemento solo se puede instalar en los **clústeres de v1.13 o anteriores**. De forma predeterminada, el complemento **everest** se instala cuando se crean clústeres de v1.15 o posterior.

### NOTA

**En un clúster de v1.13 o anteriores**, cuando una actualización o corrección de errores está disponible para las funcionalidades de almacenamiento, solo necesita instalar o actualizar el complemento del storage-driver. No es necesario actualizar el clúster ni crear un clúster.

## Instalación del complemento

Este complemento se ha instalado de forma predeterminada. Si se desinstala por alguna razón, puede volver a instalarlo realizando los siguientes pasos:

Si el controlador de almacenamiento no está instalado en un clúster, realice los siguientes pasos para instalarlo:

- Paso 1** Inicie sesión en la consola de CCE y acceda a la consola del clúster. Elija **Add-ons** en el panel de navegación, localice **storage-driver** a la derecha y haga clic en **Install**.
- Paso 2** Haga clic en **Install** para instalar el complemento. Tenga en cuenta que el storage-driver no tiene parámetros configurables y se puede instalar directamente.

---Fin

# 15 Gráfico de Helm

---

## 15.1 Descripción general

CCE proporciona una consola para la gestión de gráficos de Helm, lo que le ayuda a desplegar fácilmente aplicaciones mediante los gráficos y gestionar aplicaciones en la consola. CCE utiliza Helm v3.8.2 y soporta paquetes de gráficos Helm v3. Para obtener más información, consulte [Despliegue de una aplicación desde un gráfico](#).

También puede utilizar el cliente de Helm para desplegar aplicaciones directamente. Si utiliza el cliente de Helm para desplegar aplicaciones, no se admite el control de versiones. Puede usar Helm v2 o Helm v3. Para más detalles, véase [Despliegue de una aplicación a través del cliente de Helm v2](#) y [Despliegue de una aplicación a través del cliente de Helm v3](#).

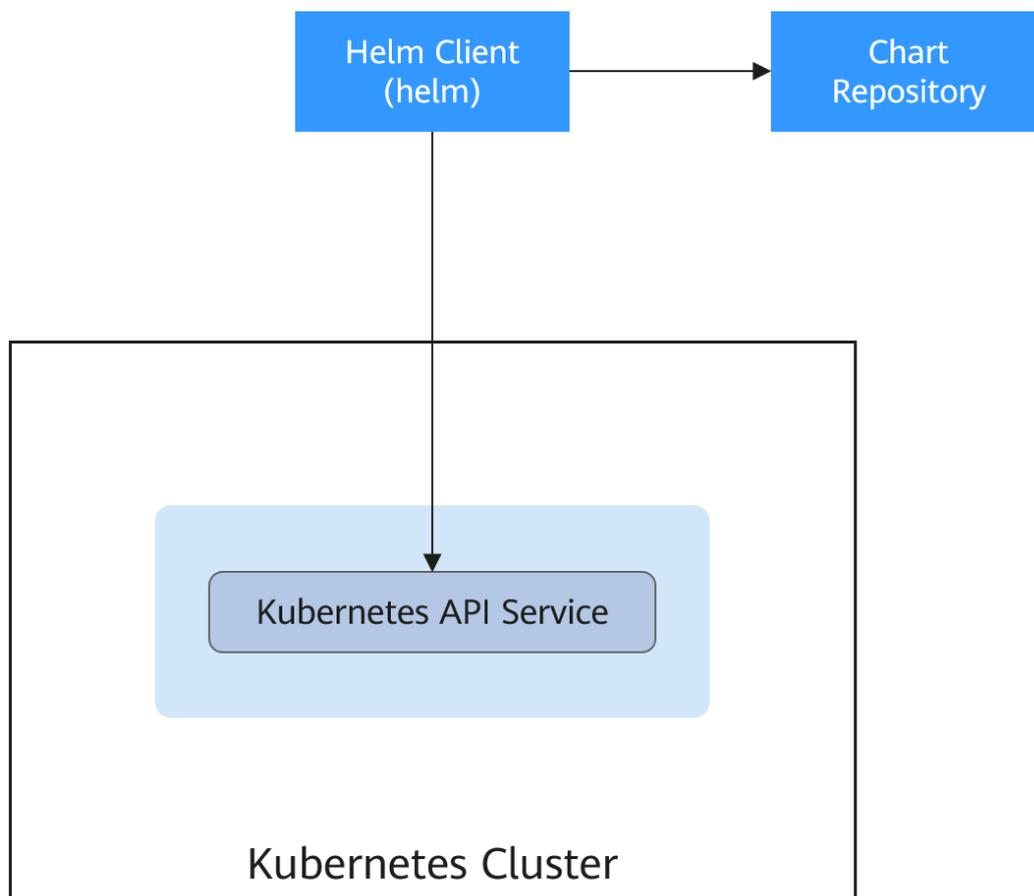
### Helm

**Helm** es un gestor de paquetes para Kubernetes y gestiona gráficos. Un gráfico de Helm es una serie de archivos YAML utilizados para encapsular aplicaciones nativas de Kubernetes. Al desplegar una aplicación, puede personalizar algunos metadatos de la aplicación para facilitar la distribución de la aplicación. Los liberadores de aplicaciones pueden usar Helm para empaquetar aplicaciones, gestionar dependencias de aplicaciones y versiones de aplicaciones, y liberar aplicaciones en el repositorio de software. Después de usar Helm, los usuarios no necesitan compilar archivos complejos de despliegue de aplicaciones. Pueden buscar, instalar, actualizar, revertir y desinstalar fácilmente aplicaciones en Kubernetes.

La relación entre Helm y Kubernetes es la siguiente:

- Helm <=> Kubernetes
- Apt <=> Ubuntu
- Yum <=> CentOS
- Pip <=> Python

La siguiente figura muestra la arquitectura de la solución:



Helm puede ayudar a orquestar aplicaciones para Kubernetes:

- Gestiona, edita y actualiza un gran número de archivos de configuración de Kubernetes.
- Despliega una aplicación compleja de Kubernetes que contiene un gran número de archivos de configuración.
- Comparte y reutiliza las configuraciones y aplicaciones de Kubernetes.
- Soporta múltiples entornos con plantillas de configuración basadas en parámetros.
- Gestiona la versión de las aplicaciones, incluida la reversión de la aplicación, la búsqueda de diferencias (con el comando **diff**) y la visualización del historial de versiones.
- Controla las fases en un ciclo despliegue.
- Prueba y verifica la versión publicada.

## 15.2 Despliegue de una aplicación desde un gráfico

En la consola de CCE, puede cargar un paquete de gráficos de Helm, desplugarlo y gestionar los pods desplegados.

## AVISO

CCE ha cambiado gradualmente a Helm v3 desde septiembre de 2022. Los gráficos de Helm v2 ya no son compatibles con la consola. Si no puede cambiar a Helm v3 por ahora, puede usar el cliente de Helm v2 para gestionar los gráficos de Helm v2 en segundo plano.

## Restricciones

- El número de gráficos que puede cargar un solo usuario es limitado. El valor que se muestra en la consola de cada región es la cantidad permitida.
- CCE utiliza Helm v3.8.2 y permite subir paquetes de gráficos de Helm v3.
- Un gráfico con varias versiones consume la misma cantidad de cuota de gráfico.
- Los usuarios con permisos de operación de gráficos pueden realizar varias operaciones en clústeres. Por lo tanto, tenga cuidado al asignar a los usuarios los permisos de gestión del ciclo de vida de los gráficos, incluida la carga de gráficos y la creación, eliminación y actualización de versiones de gráficos.

## Especificaciones de la carta

La carga de trabajo de Redis se utiliza como ejemplo para ilustrar las especificaciones del gráfico.

- **Requisito de denominación**

Un paquete de gráficos se nombra en el formato **{name}-{version}.tgz**, donde **{version}** indica el número de versión en el formato de *Major version number.Minor version number.Revision number*. Por ejemplo, **redis-0.4.2.tgz**.

### 📖 NOTA

El nombre del gráfico {nombre} puede contener un máximo de 64 caracteres.

El número de versión debe cumplir con las reglas [semánticas de control de versiones](#).

- Los números de versión principal y secundario son obligatorios, y el número de revisión es opcional.
  - Los números de versión mayor y menor y el número de revisión deben ser enteros, mayores o iguales a 0, y menores o iguales a 99.
- **Estructura de directorios**

La estructura de directorios de un gráfico es la siguiente:

```
redis/  
  templates/  
  values.yaml  
  README.md  
  Chart.yaml  
  .helmignore
```

Los parámetros marcados con \* son obligatorios según la [Tabla 15-1](#).

**Tabla 15-1** Parámetros en la estructura de directorios de un gráfico

Parámetro	Descripción
* templates	Almacena todas las plantillas.

Parámetro	Descripción
* values.yaml	<p>Describe los parámetros de configuración requeridos por las plantillas.</p> <p><b>AVISO</b></p> <p>Asegúrese de que la dirección de imagen establecida en el archivo <b>values.yaml</b> es la misma que la dirección de imagen en el repositorio de imágenes de contenedor. De lo contrario, se produce una excepción al crear una carga de trabajo y el sistema muestra un mensaje que indica que no se puede extraer la imagen.</p> <p>Para obtener la dirección de la imagen, realice las siguientes operaciones: Inicie sesión en la consola de CCE. En el panel de navegación, elija <b>Image Repository</b> para acceder a la consola de SWR. Elija <b>My Images &gt; Private Images</b> y haga clic en el nombre de la imagen cargada. En la página de ficha <b>Image Tags</b>, obtenga la dirección de imagen del comando de extracción. Puede hacer clic en  para copiar el comando en la columna <b>Image Pull Command</b>.</p>
README.md	<p>Un archivo de reducción de marca, que incluye:</p> <ul style="list-style-type: none"> <li>● La carga de trabajo o los servicios proporcionados por el gráfico.</li> <li>● Prerrequisitos para ejecutar el gráfico.</li> <li>● Configuraciones en el archivo <b>values.yaml</b>.</li> <li>● Información sobre la instalación y configuración del gráfico.</li> </ul>
* Chart.yaml	<p>Información básica sobre el gráfico.</p> <p>Nota: La versión de la API de Helm v3 se cambia de v1 a v2.</p>
.helmignore	<p>Archivos o datos que no necesitan leer plantillas durante la instalación de la carga de trabajo.</p>

## Carga de un gráfico

**Paso 1** Inicie sesión en la consola de CCE, haga clic en el nombre del clúster y acceda a la consola del clúster. Elija **Charts** en el panel de navegación y haga clic en **Upload Chart** en la esquina superior derecha.

**Paso 2** Haga clic en **Select File**, seleccione el gráfico que desea cargar y haga clic en **Upload**.

### NOTA

Al cargar un gráfico, la regla de nomenclatura del bucket de OBS cambia de `cce-charts-{region}-{domain_name}` a `cce-charts-{region}-{domain_id}`. En la antigua regla de nomenclatura, el sistema convierte el valor **domain\_name** en una cadena de Base64 y utiliza los primeros 63 caracteres. Si no puede encontrar el gráfico en el bucket de OBS con el nuevo nombre, busque el bucket con el nombre antiguo.

----Fin

## Creación de una versión

**Paso 1** Inicie sesión en la consola de CCE, haga clic en el nombre del clúster y acceda a la consola del clúster. En el panel de navegación, elija **Charts**.

**Paso 2** En la página de la ficha **My Charts**, haga clic en **Install** del gráfico de destino.

**Paso 3** Establezca los parámetros de instalación de la carga de trabajo haciendo referencia a [Tabla 15-2](#).

**Tabla 15-2** Parámetros de instalación

Parámetro	Descripción
Instance	Nombre único de la versión del gráfico.
Namespace	Espacio de nombres en el que se desplegará la carga de trabajo.
Select Version	Versión de un gráfico.
Configuration File	<p>Puede importar y reemplazar el archivo <b>values.yaml</b> o editar directamente los parámetros del gráfico en línea.</p> <p><b>NOTA</b></p> <p>Un archivo <b>values.yaml</b> importado debe cumplir con las especificaciones YAML, es decir, el formato KEY:VALUE. Los campos del archivo no están restringidos.</p> <p>El valor clave del <b>values.yaml</b> importados debe ser el mismo que el del paquete de gráficos seleccionado. De lo contrario, el <b>values.yaml</b> no tienen efecto. Es decir, la clave no se puede cambiar.</p> <ol style="list-style-type: none"> <li>Haga clic en <b>Select File</b>.</li> <li>Seleccione el archivo <b>values.yaml</b> correspondiente y haga clic en <b>Open</b>.</li> </ol>

**Paso 4** Haga clic en **Install**.

En la página de ficha **Releases**, puede ver el estado de instalación de la versión.

----Fin

## Actualización de una carga de trabajo basada en gráficos

**Paso 1** Inicie sesión en la consola de CCE, haga clic en el nombre del clúster y acceda a la consola del clúster. Elija **Charts** en el panel de navegación y haga clic en la ficha **Releases**.

**Paso 2** Haga clic en **Upgrade** en la fila donde reside la carga de trabajo deseada y establezca los parámetros para la carga de trabajo.

**Paso 3** Seleccione una versión de gráfico para **Chart Version**.

**Paso 4** Siga las indicaciones para modificar los parámetros del gráfico. Haga clic en **Upgrade** y, a continuación, haga clic en **Submit**.

**Paso 5** Haga clic en **Back to Release List**. Si el estado del gráfico cambia a **Upgrade successful**, la carga de trabajo se actualiza correctamente.

----Fin

## Revertir una carga de trabajo basada en gráficos

**Paso 1** Inicie sesión en la consola de CCE, haga clic en el nombre del clúster y acceda a la consola del clúster. Elija **Charts** en el panel de navegación y haga clic en la ficha **Releases**.

**Paso 2** Haga clic en **More > Roll Back** para la carga de trabajo que se va a revertir, seleccione la versión de la carga de trabajo y haga clic en **Roll back to this version**.

En la lista de carga de trabajo, si el estado es **Rollback successful**, la carga de trabajo se revierte correctamente.

----Fin

## Desinstalación de una carga de trabajo basada en gráficos

**Paso 1** Inicie sesión en la consola de CCE, haga clic en el nombre del clúster y acceda a la consola del clúster. Elija **Charts** en el panel de navegación y haga clic en la ficha **Releases**.

**Paso 2** Haga clic en **More > Uninstall** junto a la versión que se va a desinstalar y haga clic en **Yes**. Tenga cuidado al realizar esta operación porque las versiones no se pueden restaurar después de desinstalarlas.

----Fin

## 15.3 Diferencias entre Helm v2 y Helm v3 y soluciones de adaptación

Helm v2 se detiene en la versión 2.17.0. Actualmente, Helm v3 es el estándar en la comunidad Helm. Se le aconseja cambiar sus gráficos al **formato de Helm v3** tan pronto como sea posible.

Cambios desde Helm v2:

### 1. Remoción de timón

Helm v3 es más simple y más fácil de usar. Quita el timón y se conecta directamente al servidor de API usando kubeconfig, simplificando el modelo de seguridad.

### 2. Estrategia de actualización mejorada: parches de fusión estratégica de 3 vías

Helm v2 usó un parche de fusión estratégica bidireccional. Durante una actualización, comparó el manifiesto del gráfico más reciente con el manifiesto del gráfico propuesto para determinar qué cambios debían aplicarse a los recursos en Kubernetes. Si se aplicaron cambios al clúster fuera de banda (como durante una edición de kubectl), esos cambios no se consideraron. Esto dio lugar a que los recursos no pudieran volver a su estado anterior.

Helm v3 utiliza un parche de fusión estratégica de tres vías. Helm considera el viejo manifiesto, su estado vivo, y el nuevo manifiesto cuando genera un parche. Helm compara el estado activo actual con el estado activo del manifiesto antiguo, comprueba si se modifica el nuevo manifiesto y complementa automáticamente el nuevo manifiesto para generar el parche final de actualización.

Para obtener más detalles y ejemplos, consulte [https://v3.helm.sh/docs/faq/changes\\_since\\_helm2](https://v3.helm.sh/docs/faq/changes_since_helm2).

### 3. Secretos como controlador de almacenamiento predeterminado

Helm v2 usaba ConfigMaps de forma predeterminada para almacenar información de lanzamiento. En Helm v3, los secretos ahora se utilizan como el controlador de almacenamiento predeterminado.

### 4. Los nombres de versiones ahora están en el ámbito del espacio de nombres

En Helm v2, la información sobre cada versión se almacenaba en el mismo espacio de nombres que Tiller. En la práctica, esto significaba que una vez que un nombre fue utilizado por una versión, ninguna otra versión podría usar ese mismo nombre, incluso si se desplegó en un espacio de nombres diferente. En Helm v3, la información sobre una versión en particular se almacena ahora en el mismo espacio de nombres que la versión en sí. Esto significa que el nombre de la versión se puede usar en diferentes espacios de nombres. El espacio de nombres de la aplicación es el mismo que el de la versión.

#### 5. Cambio de modo de verificación

Helm v3 verifica el formato del gráfico más estrictamente. Por ejemplo, Helm v3 golpea la `apiVersion` en `Chart.yaml` de v1 a v2. Para el `Chart.yaml` de v2, `apiVersion` debe establecerse en v1. Después de instalar el cliente Helm v3, puede ejecutar el comando **helm lint** para comprobar si el formato del gráfico cumple con las especificaciones de Helm v3.

**Solución de adaptación:** Adapta el gráfico de Helm v3 basado en el documento oficial de Helm <https://helm.sh/docs/topics/charts/>. El campo `apiVersion` es obligatorio.

#### 6. Eliminación del gancho `crd-install`

El gancho `crd-install` se ha eliminado a favor del directorio `crds/` en Helm v3. Tenga en cuenta que los recursos del directorio `crds/` solo se despliegan durante la instalación de la versión y no se actualizan durante la actualización. Cuando se eliminan los recursos, los recursos se conservan en el directorio `crds/`. Si el CRD ya existe, se omitirá con una advertencia durante la instalación repetida.

**Solución de adaptación:** De acuerdo con el [documento de Helm](#), puede mantener su CRD en el directorio `crds/` o en un gráfico separado. Helm no puede actualizar o eliminar el CRD. Por lo tanto, se recomienda poner el CRD en un gráfico y, a continuación, poner los recursos que utilizan ese CRD en otro gráfico.

#### 7. Los recursos que no se crean con Helm no se actualizan a la fuerza. Las versiones no se actualizan a la fuerza de forma predeterminada.

Se cambia la lógica de actualización forzada de Helm v3. Después de que la actualización falla, el sistema no elimina y reconstruye el Helm v3. En su lugar, el sistema utiliza directamente la lógica `put`. Por lo tanto, la actualización de la versión de CCE utiliza la lógica de actualización no forzada de forma predeterminada. Los recursos que no se pueden actualizar mediante parches harán que la versión no se pueda actualizar. Si existe una versión con el mismo nombre en el entorno y no tiene la etiqueta de inicio `app.kubernetes.io/managed-by: Helm` de Helm v3, se muestra un mensaje de conflicto.

**Solución de adaptación:** Eliminar recursos relacionados y crearlos usando Helm.

#### 8. Límite en los registros históricos de liberación

De forma predeterminada, solo se conservan las 10 versiones más recientes.

**Para más cambios y detalles, consulte los documentos oficiales de Helm.**

- Diferencias entre Helm v2 y Helm v3: [https://v3.helm.sh/docs/faq/changes\\_since\\_helm2](https://v3.helm.sh/docs/faq/changes_since_helm2)
- Cómo migrar de Helm v2 a Helm v3: [https://helm.sh/docs/topics/v2\\_v3\\_migration](https://helm.sh/docs/topics/v2_v3_migration)

## 15.4 Despliegue de una aplicación a través del cliente de Helm v2

### AVISO

CCE ha cambiado gradualmente a Helm v3 desde septiembre de 2022. Los gráficos de Helm v2 ya no son compatibles con la consola. Si no puede cambiar a Helm v3 por ahora, puede usar el cliente de Helm v2 para gestionar los gráficos de Helm v2 en segundo plano.

### Requisitos previos

El clúster de Kubernetes creado en CCE se ha conectado a `kubectl`. Para obtener más información, véase [Uso de kubectl](#).

### Precauciones

CCE intentará convertir las versiones v2 a v3. Si elimina una versión de Helm v2 en segundo plano, la información de la versión todavía se muestra en la página de gráficos de la consola de CCE. En este caso, elimínelo.

### Instalación de Helm v2

Esta sección utiliza Helm v2.17.0 como ejemplo.

Para otras versiones, visite <https://github.com/helm/helm/releases>.

**Paso 1** Descargue el cliente Helm desde la máquina virtual conectada al clúster.

```
wget https://get.helm.sh/helm-v2.17.0-linux-amd64.tar.gz
```

**Paso 2** Descomprima el paquete Helm.

```
tar -xzf helm-v2.17.0-linux-amd64.tar.gz
```

**Paso 3** Copie Helm en la ruta del sistema, por ejemplo, `/usr/local/bin/helm`.

```
mv linux-amd64/helm /usr/local/bin/helm
```

**Paso 4** RBAC está habilitado en el servidor de API de Kubernetes. Por lo tanto, debe crear el nombre de cuenta de servicio `tiller` para el timón y asignar `cluster-admin`, un `ClusterRole` de sistema, al timón. Cree una cuenta de recursos de timón de la siguiente manera:

#### vim tiller-rbac.yaml

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: tiller
  namespace: kube-system
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: tiller
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
```

```
subjects:
- kind: ServiceAccount
  name: tiller
  namespace: kube-system
```

**Paso 5** Despliegue la cuenta de recursos de timón.

```
kubectl apply -f tiller-rbac.yaml
```

**Paso 6** Inicialice el Helm y despliegue el pod del timón.

```
helm init --service-account tiller --skip-refresh
```

**Paso 7** Consulte el estado.

```
kubectl get pod -n kube-system -l app=helm
```

Salida del comando:

NAME	READY	STATUS	RESTARTS	AGE
tiller-deploy-7b56c8dfb7-fxk5g	1/1	Running	1	23h

**Paso 8** Consulte la versión Helm.

```
# helm version
Client: &version.Version{SemVer:"v2.17.0",
GitCommit:"a690bad98af45b015bd3dala41f6218b1a451dbe", GitTreeState:"clean"}
Server: &version.Version{SemVer:"v2.17.0",
GitCommit:"a690bad98af45b015bd3dala41f6218b1a451dbe", GitTreeState:"clean"}
```

----Fin

## Instalación del gráfico Helm

Si los gráficos proporcionados por CCE no cumplen con los requisitos, descargue un gráfico e instálelo.

Puede obtener el gráfico requerido en el directorio **stable** de este sitio web de <https://github.com/helm/charts> y descargarlo y subirlo al nodo.

1. Descargue y descomprima el gráfico obtenido. Generalmente, el gráfico está en formato ZIP.  

```
unzip chart.zip
```
2. Instale el gráfico de Helm.  

```
helm install aerospike/
```
3. Una vez completada la instalación, ejecute el comando **helm list** para comprobar el estado de las versiones de gráficos.

## Problemas comunes

- Aparece el siguiente mensaje de error después de ejecutar el comando **helm version**:

```
Client:
&version.Version{SemVer:"v2.17.0",
GitCommit:"a690bad98af45b015bd3dala41f6218b1a451dbe", GitTreeState:"clean"}
E0718 11:46:10.132102 7023 portforward.go:332] an error occurred
forwarding 41458 -> 44134: error forwarding port 44134 to pod
d566b78f997eea6c4b1c0322b34ce8052c6c2001e8edff243647748464cd7919, uid : unable
to do port forwarding: socat not found.
Error: cannot connect to Tiller
```

La información anterior se muestra porque el socat no está instalado. Ejecute el siguiente comando para instalar el socat:

```
yum install socat -y
```

- Cuando ejecuta el comando **yum install socat -y** en un nodo que ejecuta EulerOS 2.9 y se muestra el siguiente mensaje de error:

No hay coincidencia para el argumento: socat

Error: no se puede encontrar una coincidencia: socat

Esto indica que la imagen de socat no está contenida. Descargue el paquete RPM de [https://repo.huaweicloud.com/openeuler/openEuler-20.03-LTS-SP3/everything/x86\\_64/Packages/socat-1.7.3.2-8.oe1.x86\\_64.rpm](https://repo.huaweicloud.com/openeuler/openEuler-20.03-LTS-SP3/everything/x86_64/Packages/socat-1.7.3.2-8.oe1.x86_64.rpm) y ejecute el siguiente comando para instalar socat:

```
rpm -i socat-1.7.3.2-8.oe1.x86_64.rpm
```

- Cuando se ha instalado el socat y se muestra el siguiente mensaje de error después de ejecutar el comando **helm version**:

```
test@local:~/k8s/helm/test$ helm version
Client: &version.Version{SemVer:"v3.3.0",
GitCommit:"021cb0acla1b2f888144ef5a67b8dab6c2d45be6", GitTreeState:"clean"}
Error: cannot connect to Tiller
```

El gráfico Helm lee el certificado de configuración del archivo **.Kube/config** para comunicarse con Kubernetes. El error anterior indica que la configuración de kubectl es incorrecta. En este caso, vuelva a conectar el clúster a kubectl. Para obtener más información, véase [Uso de kubectl](#).

- El almacenamiento no se crea después de que se haya conectado a los servicios de almacenamiento en la nube.

Este problema puede ser causado por el campo **annotation** en el PVC creado. Cambie el nombre del gráfico e instale el gráfico de nuevo.

- Si kubectl no está configurado correctamente, se muestra el siguiente mensaje de error después de ejecutar el comando **helm install**:

```
[root@prometheus-57046 ~]# helm install prometheus/ --generate-name
WARNING: This chart is deprecated
Error: Kubernetes cluster unreachable: Get "http://localhost:8080/version?
timeout=32s": dial tcp [::1]:8080: connect: connection refused
```

**Solución:** Configurar kubeconfig para el nodo. Para obtener más información, véase [Uso de kubectl](#).

## 15.5 Despliegue de una aplicación a través del cliente de Helm v3

### Requisitos previos

El clúster de Kubernetes creado en CCE se ha conectado a kubectl. Para obtener más información, véase [Uso de kubectl](#).

### Instalación de Helm v3

Esta sección usa Helm v3.3.0 como ejemplo.

Para otras versiones, visite <https://github.com/helm/helm/releases>.

**Paso 1** Descargue el cliente Helm desde la máquina virtual conectada al clúster.

```
wget https://get.helm.sh/helm-v3.3.0-linux-amd64.tar.gz
```

**Paso 2** Descomprima el paquete Helm.

```
tar -xzvf helm-v3.3.0-linux-amd64.tar.gz
```

**Paso 3** Copie Helm en la ruta del sistema, por ejemplo, **/usr/local/bin/helm**.

```
mv linux-amd64/helm /usr/local/bin/helm
```

**Paso 4** Consulte la versión Helm.

```
helm version
version.BuildInfo{Version:"v3.3.0",
GitCommit:"e29ce2a54e96cd02ccf88bee4f58bb6e2a28b6", GitTreeState:"clean",
GoVersion:"go1.13.4"}
```

----Fin

## Instalación del gráfico Helm

Si los gráficos proporcionados por CCE no cumplen con los requisitos, descargue un gráfico e instálelo.

Puede obtener el gráfico requerido en el directorio **stable** de este sitio web de <https://github.com/helm/charts> y descargarlo y subirlo al nodo.

1. Descargue y descomprima el gráfico obtenido. Generalmente, el gráfico está en formato ZIP.

```
unzip chart.zip
```

2. Instale el gráfico de Helm.

```
helm install aerospike/ --generate-name
```

3. Una vez completada la instalación, ejecute el comando **helm list** para comprobar el estado de las versiones de gráficos.

## Problemas comunes

- Aparece el siguiente mensaje de error después de ejecutar el comando **helm version**:

```
Client:
&version.Version{SemVer:"v3.3.0",
GitCommit:"012cb0acla1b2f888144ef5a67b8dab6c2d45be6", GitTreeState:"clean"}
E0718 11:46:10.132102 7023 portforward.go:332] an error occurred
forwarding 41458 -> 44134: error forwarding port 44134 to pod
d566b78f997eea6c4b1c0322b34ce8052c6c2001e8edff243647748464cd7919, uid : unable
to do port forwarding: socat not found.
Error: cannot connect to Tiller
```

La información anterior se muestra porque el socat no está instalado. Ejecute el siguiente comando para instalar el socat:

```
yum install socat -y
```

- Cuando ejecuta el comando **yum install socat -y** en un nodo que ejecuta EulerOS 2.9 y se muestra el siguiente mensaje de error:

No hay coincidencia para el argumento: socat

Error: no se puede encontrar una coincidencia: socat

Esto indica que la imagen de socat no está contenida. Descargue el paquete RPM de [https://repo.huaweicloud.com/openeuler/openEuler-20.03-LTS-SP3/everything/x86\\_64/Packages/socat-1.7.3.2-8.oe1.x86\\_64.rpm](https://repo.huaweicloud.com/openeuler/openEuler-20.03-LTS-SP3/everything/x86_64/Packages/socat-1.7.3.2-8.oe1.x86_64.rpm) y ejecute el siguiente comando para instalar socat:

```
rpm -i socat-1.7.3.2-8.oe1.x86_64.rpm
```

- Cuando se ha instalado el socat y se muestra el siguiente mensaje de error después de ejecutar el comando **helm version**:

```
$ helm version
Client: &version.Version{SemVer:"v3.3.0",
GitCommit:"021cb0acla1b2f888144ef5a67b8dab6c2d45be6", GitTreeState:"clean"}
Error: cannot connect to Tiller
```

El gráfico Helm lee el certificado de configuración del archivo **.Kube/config** para comunicarse con Kubernetes. El error anterior indica que la configuración de kubectl es

incorrecta. En este caso, vuelva a conectar el clúster a kubectl. Para obtener más información, véase [Uso de kubectl](#).

- El almacenamiento no se crea después de que se haya conectado a los servicios de almacenamiento en la nube.

Este problema puede ser causado por el campo **annotation** en el PVC creado. Cambie el nombre del gráfico e instale el gráfico de nuevo.

- Si kubectl no está configurado correctamente, se muestra el siguiente mensaje de error después de ejecutar el comando **helm install**:

```
# helm install prometheus/ --generate-name
WARNING: This chart is deprecated
Error: Kubernetes cluster unreachable: Get "http://localhost:8080/version?
timeout=32s": dial tcp [::1]:8080: connect: connection refused
```

**Solución:** Configurar kubeconfig para el nodo. Para obtener más información, véase [Uso de kubectl](#).

## 15.6 Convertir una versión de Helm v2 a v3

### Contexto

CCE es totalmente compatible con Helm v3. Esta sección le guía para convertir una versión de Helm v2 a Helm v3. Helm v3 descarta o reconstruye algunas funciones Helm v2 en la capa inferior. Por lo tanto, la conversión es arriesgada hasta cierto punto. Se requiere simulación antes de la conversión.

Para obtener más información, consulte la [documentación de la comunidad](#).

### Precauciones

- Helm v2 almacena información de lanzamiento en el ConfigMaps. Helm v3 lo hace en secreto.
- Cuando consulta, actualiza u opera una versión Helm v2 en la consola de CCE, CCE intentará convertir la versión a v3. Si opera en segundo plano, convierta la versión siguiendo las instrucciones que aparecen a continuación.

### Proceso de conversión (sin usar el cliente Helm v3)

**Paso 1** Descargue el complemento de conversión helm 2to3 en el nodo CCE.

```
wget https://github.com/helm/helm-2to3/releases/download/v0.10.2/
helm-2to3_0.10.2_linux_amd64.tar.gz
```

**Paso 2** Descomprime el paquete de complementos.

```
tar -xzf helm-2to3_0.10.2_linux_amd64.tar.gz
```

**Paso 3** Realice la conversión simulada.

Tome la versión test-convert como ejemplo. Ejecute el siguiente comando para simular la conversión: Si se muestra la siguiente información, la simulación se realiza correctamente.

```
# ./2to3 convert --dry-run --tiller-out-cluster -s configmaps test-convert
NOTE: This is in dry-run mode, the following actions will not be executed.
Run without --dry-run to take the actions described below:
Release "test-convert" will be converted from Helm v2 to Helm v3.
[Helm 3] Release "test-convert" will be created.
[Helm 3] ReleaseVersion "test-convert.v1" will be created.
```

**Paso 4** Realice la conversión. Si se muestra la siguiente información, la conversión se realiza correctamente.

```
# ./2to3 convert --tiller-out-cluster -s configmaps test-convert
Release "test-convert" will be converted from Helm v2 to Helm v3.
[Helm 3] Release "test-convert" will be created.
[Helm 3] ReleaseVersion "test-convert.v1" will be created.
[Helm 3] ReleaseVersion "test-convert.v1" created.
[Helm 3] Release "test-convert" created.
Release "test-convert" was converted successfully from Helm v2 to Helm v3.
Note: The v2 release information still remains and should be removed to avoid
conflicts with the migrated v3 release.
v2 release information should only be removed using `helm 2to3` cleanup and when
all releases have been migrated over.
```

**Paso 5** Una vez completada la conversión, simule la liquidación de recursos. Después de la simulación, borre los recursos de la versión v2.

Espacio libre simulado:

```
# ./2to3 cleanup --dry-run --tiller-out-cluster -s configmaps --name test-convert
NOTE: This is in dry-run mode, the following actions will not be executed.
Run without --dry-run to take the actions described below:
WARNING: "Release 'test-convert' Data" will be removed.
[Cleanup/confirm] Are you sure you want to cleanup Helm v2 data? [y/N]: y
Helm v2 data will be cleaned up.
[Helm 2] Release 'test-convert' will be deleted.
[Helm 2] ReleaseVersion "test-convert.v1" will be deleted.
```

Espacio libre formal:

```
# ./2to3 cleanup --tiller-out-cluster -s configmaps --name test-convert
WARNING: "Release 'test-convert' Data" will be removed.
[Cleanup/confirm] Are you sure you want to cleanup Helm v2 data? [y/N]: y
Helm v2 data will be cleaned up.
[Helm 2] Release 'test-convert' will be deleted.
[Helm 2] ReleaseVersion "test-convert.v1" will be deleted.
[Helm 2] ReleaseVersion "test-convert.v1" d
```

----Fin

## Proceso de conversión (con el cliente de Helm v3)

**Paso 1** Instale el cliente de Helm v3. Para obtener más información, véase [Instalación de Helm v3](#).

**Paso 2** Instale el complemento de conversión.

```
# helm plugin install https://github.com/helm/helm-2to3
Downloading and installing helm-2to3 v0.10.2 ...
https://github.com/helm/helm-2to3/releases/download/v0.10.2/
helm-2to3_0.10.2_linux_amd64.tar.gz
Installed plugin: 2to3
```

**Paso 3** Compruebe si el complemento ha sido instalado.

```
# helm plugin list
NAME VERSION
DESCRIPTION
2to3 0.10.2 migrate and cleanup Helm v2 configuration and releases in-place
to Helm v3
```

**Paso 4** Realice la conversión simulada.

Tome la versión test-convert como ejemplo. Ejecute el siguiente comando para simular la conversión: Si se muestra la siguiente información, la conversión simulada se realiza correctamente.

```
# helm 2to3 convert --dry-run --tiller-out-cluster -s configmaps test-convert
NOTE: This is in dry-run mode, the following actions will not be executed.
```

```
Run without --dry-run to take the actions described below:
Release "test-convert" will be converted from Helm v2 to Helm v3.
[Helm 3] Release "test-convert" will be created.
[Helm 3] ReleaseVersion "test-convert.v1" will be created.
```

**Paso 5** Realice la conversión. Si se muestra la siguiente información, la conversión se realiza correctamente.

```
# helm 2to3 convert --tiller-out-cluster -s configmaps test-convert
Release "test-convert" will be converted from Helm v2 to Helm v3.
[Helm 3] Release "test-convert" will be created.
[Helm 3] ReleaseVersion "test-convert.v1" will be created.
[Helm 3] ReleaseVersion "test-convert.v1" created.
[Helm 3] Release "test-convert" created.
Release "test-convert" was converted successfully from Helm v2 to Helm v3.
Note: The v2 release information still remains and should be removed to avoid
conflicts with the migrated v3 release.
v2 release information should only be removed using `helm 2to3` cleanup and when
all releases have been migrated over.
```

**Paso 6** Después de la conversión, puede ver la versión convertida ejecutando **helm list**.

```
# helm list
NAME                NAMESPACE      REVISION UPDATED
STATUS             CHART          APP VERSION
test-convert       default         1          2022-08-29 06:56:28.166918487 +0000
UTC                deployed      test-helmold-1
```

**Paso 7** Una vez completada la conversión, simule la liquidación de recursos. Después de la simulación, borre los recursos de la versión v2.

Espacio libre simulado:

```
# helm 2to3 cleanup --dry-run --tiller-out-cluster -s configmaps --name test-convert
NOTE: This is in dry-run mode, the following actions will not be executed.
Run without --dry-run to take the actions described below:
WARNING: "Release 'test-convert' Data" will be removed.
[Cleanup/confirm] Are you sure you want to cleanup Helm v2 data? [y/N]: y
Helm v2 data will be cleaned up.
[Helm 2] Release 'test-convert' will be deleted.
[Helm 2] ReleaseVersion "test-convert.v1" will be deleted.
```

Espacio libre formal:

```
# helm 2to3 cleanup --tiller-out-cluster -s configmaps --name test-convert
WARNING: "Release 'test-convert' Data" will be removed.
[Cleanup/confirm] Are you sure you want to cleanup Helm v2 data? [y/N]: y
Helm v2 data will be cleaned up.
[Helm 2] Release 'test-convert' will be deleted.
[Helm 2] ReleaseVersion "test-convert.v1" will be deleted.
[Helm 2] ReleaseVersion "test-convert.v1" deleted.
[Helm 2] Release 'test-convert' deleted.
Helm v2 data was cleaned up successfully.
```

----Fin

# 16 Permisos

---

## 16.1 Descripción de permisos

La gestión de permisos de CCE le permite asignar permisos a los usuarios y grupos de usuarios de IAM en sus cuentas de tenant. CCE combina las ventajas de Identity and Access Management (IAM) y Kubernetes Role-based Access Control (RBAC) para proporcionar una variedad de métodos de autorización, incluida la autorización de grano fino de IAM, la autorización de token de IAM, autorización en el ámbito del clúster y autorización en todo el espacio de nombres.

CCE le permite gestionar los permisos en clústeres y recursos relacionados con una granularidad más fina, por ejemplo, para controlar el acceso de los empleados de diferentes departamentos a los recursos en la nube.

Esta sección describe el mecanismo de gestión de permisos de CCE y conceptos relacionados. Si su cuenta ha cumplido con sus requisitos de servicio, puede omitir las configuraciones de este capítulo.

### Gestión de permisos de CCE

Los permisos de CCE se describen de las siguientes maneras:

- **Cluster-level permissions:** La gestión de permisos a nivel de clúster evoluciona de la función de autorización de políticas del sistema de IAM. Los usuarios de IAM del mismo grupo de usuarios tienen los mismos permisos. En IAM, puede configurar las políticas del sistema para describir qué grupos de usuarios de IAM pueden realizar las operaciones en los recursos del clúster. Por ejemplo, puede conceder al grupo de usuarios A que cree y elimine el clúster X, agregue un nodo o instale un complemento, mientras que concede al grupo de usuarios B que vea información sobre el clúster X.

Los permisos a nivel de clúster implican API de CCE que no son de Kubernetes y admiten políticas de IAM detalladas y la gestión de proyectos empresariales.

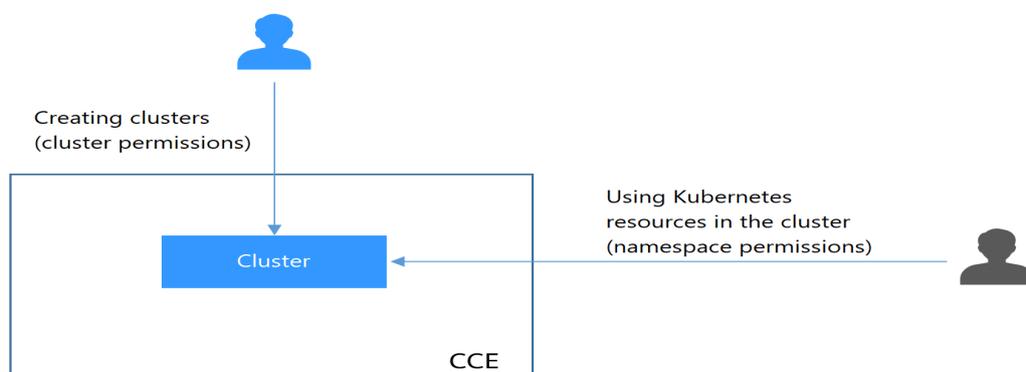
- **Namespace-level permissions:** Puede regular el acceso de los usuarios o de grupos de usuarios a los recursos de Kubernetes en un solo espacio de nombres en función de sus roles de RBAC de Kubernetes. CCE también se ha mejorado sobre la base de las capacidades de código abierto. Admite la autorización de RBAC basada en el usuario o grupo de usuarios de IAM, y la autenticación de RBAC en el acceso a las API mediante tokens de IAM.

Los permisos a nivel de espacio de nombres implican API de CCE Kubernetes y se mejoran en función de las capacidades RBAC de Kubernetes. Los permisos de nivel de espacio de nombres se pueden conceder a los usuarios o grupos de usuarios de IAM para la autenticación y la autorización, pero son independientes de las políticas de IAM detalladas.

A partir de la versión 1.11.7-r2, los clústeres de CCE permiten configurar permisos de espacio de nombres. Los clústeres anteriores a v1.11.7-r2 tienen todos los permisos de espacio de nombres de forma predeterminada.

En general, se configuran los permisos de CCE en dos escenarios. El primero es crear y gestionar clústeres y recursos relacionados, como nodos. El segundo es crear y usar recursos de Kubernetes en el clúster, como cargas de trabajo y Servicios.

**Figura 16-1** Ilustración sobre permisos de CCE



Estos permisos le permiten gestionar los usuarios de recursos con una granularidad más precisa.

## Cluster Permissions (basados en IAM) y Namespace Permissions (basados en Kubernetes RBAC)

Los usuarios con diferentes permisos de clúster (asignados mediante IAM) tienen diferentes permisos de espacio de nombres (asignados mediante Kubernetes RBAC). [Tabla 16-1](#) enumera los permisos de espacio de nombres de diferentes usuarios.

**Tabla 16-1** Diferencias en los permisos de espacio de nombres

Usuario	Clústeres de v1.13 y posteriores
Usuario con los permisos de Tenant Administrator (por ejemplo, una cuenta)	Todos los permisos de espacio de nombres
Usuario de IAM con el rol CCE Administrator	Todos los permisos de espacio de nombres
Usuario de IAM con el rol CCE FullAccess o CCE ReadOnlyAccess	Requiere la autorización RBAC de Kubernetes.
Usuario de IAM con el rol Tenant Guest	Requiere la autorización RBAC de Kubernetes.

## Permisos de kubectl

Puede usar **kubectl** para acceder a los recursos de Kubernetes en un clúster.

Cuando se accede a un clúster mediante kubectl, CCE utiliza el archivo kubeconfig.json generado en el clúster para la autenticación. Este archivo contiene información del usuario, basada en la cual CCE determina qué recursos de Kubernetes puede acceder kubectl. Los permisos registrados en un archivo kubeconfig.json varían de usuario a usuario. **Tabla 16-1** muestra los permisos que tiene un usuario.

## Usuarios federados

IAM proporciona la función de proveedor de identidad para implementar la autenticación de identidad federada basada en Security Assertion Markup Language (SAML) o OpenID Connect. Esta función permite a los usuarios de su sistema de gestión acceder a la plataforma en la nube a través del inicio de sesión único (SSO).

Los usuarios que inician sesión con la autenticación de identidad federada se denominan los usuarios federados. Los usuarios federados son equivalentes a los usuarios de IAM.

Preste atención a lo siguiente para que los usuarios federados usen CCE:

- Cuando un usuario crea un clúster de CCE, el permiso de cluster-admin se concede al usuario de forma predeterminada. El ID de usuario de un usuario federado cambia en cada inicio de sesión y cierre de sesión. Por lo tanto, el usuario se muestra como eliminado en la página **Permissions** de la consola de CCE. No elimine manualmente el permiso, de lo contrario, la autenticación falla. En este caso, se recomienda conceder el permiso de cluster-admin a un grupo de usuarios en CCE y agregar los usuarios federados al grupo de usuarios.
- Los usuarios federados no pueden crear claves de acceso permanentes (AKs/SKs). En escenarios en los que se requieren AK/SK (por ejemplo, al crear PV/PVC relacionados con OBS), solo usted o un usuario de IAM puede crear AK/SK y compartirlos con los usuarios federados. Una clave de acceso contiene los permisos concedidos a un usuario, por lo que se recomienda que el usuario federado solicite a un usuario IAM del mismo grupo que cree una clave de acceso.

## Acciones admitidas

Hay dos tipos de políticas: las definidas por el sistema y las personalizadas. Si los permisos preestablecidos en el sistema no cumplen con sus requisitos, puede crear las políticas personalizadas y aplicarlas a los grupos de usuarios para un control de acceso refinado. Las operaciones admitidas por las políticas son específicas de las API. Los siguientes son conceptos comunes relacionados con las políticas:

- Permiso: una declaración en una política que permite o niega ciertas operaciones.
- API: las API de REST que se pueden invocar en una política personalizada.
- Acciones: agregadas a una política personalizada para controlar los permisos para operaciones específicas.
- Acciones dependientes: acciones de las que depende una acción específica para tener efecto. Al asignar permisos para la acción a un usuario, también debe asignar permisos para las acciones dependientes.
- Proyectos de IAM y proyectos de empresa: tipo de proyectos para los que una acción entrará en vigor. Las políticas que contienen las acciones que admiten los proyectos de IAM y de empresa se pueden asignar a los grupos de usuarios y tener efecto tanto en

IAM como en Enterprise Management. Las políticas que solo contienen las acciones que admiten los proyectos de IAM se pueden asignar a los grupos de usuarios y solo tienen efecto para IAM. Dichas políticas no entrarán en vigor si se asignan a los grupos de usuarios en Enterprise Management. Para obtener más información, consulte [¿Cuáles son las diferencias entre IAM y Enterprise Management?](#)

 **NOTA**

La marca de verificación (√) y el símbolo de cruz (x) indican respectivamente que una acción tiene efecto o no tiene efecto para el tipo correspondiente de proyectos.

CCE admite las siguientes acciones que se pueden definir en las políticas personalizadas:

**Tabla 16-2** Cluster

Permiso	API	Acción	Proyectos de IAM	Proyecto empresarial
Listado de clústeres en un proyecto especificado	GET /api/v3/projects/{project_id}/clusters	cce:cluster:list	√	√
Obtención de información acerca de un clúster especificado	GET /api/v3/projects/{project_id}/clusters/{cluster_id}	cce:cluster:get	√	√
Creación de un clúster	POST /api/v3/projects/{project_id}/clusters	cce:cluster:create	√	√
Actualización de información acerca de un clúster especificado	PUT /api/v3/projects/{project_id}/clusters/{cluster_id}	cce:cluster:update	√	√
Eliminación de un clúster	DELETE /api/v3/projects/{project_id}/clusters/{cluster_id}	cce:cluster:delete	√	√
Despierta de un clúster	POST /api/v3/projects/{project_id}/clusters/{cluster_id}/operation/awake	cce:cluster:start	√	√
Hibernación de un clúster	POST /api/v3/projects/{project_id}/clusters/{cluster_id}/operation/hibernate	cce:cluster:stop	√	√

Permiso	API	Acción	Proyectos de IAM	Proyecto empresarial
Obtención de un certificado de clúster	POST /api/v3/projects/{project_id}/clusters/{cluster_id}/clustercert	cce:cluster:get	√	√

**Tabla 16-3** Node

Permiso	API	Acción	Proyectos de IAM	Proyecto empresarial
Obtención de información acerca de todos los nodos de un clúster	GET /api/v3/projects/{project_id}/clusters/{cluster_id}/nodes	cce:node:list	√	√
Obtención de información acerca de un nodo especificado	GET /api/v3/projects/{project_id}/clusters/{cluster_id}/nodes/{node_id}	cce:node:get	√	√
Creación de un nodo	POST /api/v3/projects/{project_id}/clusters/{cluster_id}/nodes	cce:node:create	√	√
Actualización de información acerca de un nodo especificado	PUT /api/v3/projects/{project_id}/clusters/{cluster_id}/nodes/{node_id}	cce:node:update	√	√
Eliminación de un nodo	DELETE /api/v3/projects/{project_id}/clusters/{cluster_id}/nodes/{node_id}	cce:node:delete	√	√

**Tabla 16-4** Job

Permiso	API	Acción	Proyectos de IAM	Proyecto empresarial
Obtención de información sobre un trabajo	GET /api/v3/projects/{project_id}/jobs/{job_id}	cce:job:get	√	√

**Tabla 16-5** Nodepool

Permiso	API	Acción	Proyectos de IAM	Proyecto empresarial
Obtención de información sobre todos los pools de nodos de un clúster especificado	GET /api/v3/projects/{project_id}/clusters/{cluster_id}/nodepools	cce:nodepool:list	√	√
Obtención de información sobre un pool de nodo	GET /api/v3/projects/{project_id}/clusters/{cluster_id}/nodepools/{nodepool_id}	cce:nodepool:get	√	√
Creación de un pool de nodos	POST /api/v3/projects/{project_id}/clusters/{cluster_id}/nodepools	cce:nodepool:create	√	√
Actualización de información sobre un pool de nodos	PUT /api/v3/projects/{project_id}/clusters/{cluster_id}/nodepools/{nodepool_id}	cce:nodepool:update	√	√
Eliminación de un pool de nodo	DELETE /api/v3/projects/{project_id}/clusters/{cluster_id}/nodepools/{nodepool_id}	cce:nodepool:delete	√	√

**Tabla 16-6 Storage**

Permiso	API	Acción	Proyectos de IAM	Proyecto empresarial
Creación de un PersistentVolumeClaim	POST /api/v1/namespaces/{namespace}/cloudpersistentvolumeclaims	cce:storage:create	√	√
Eliminación de un PersistentVolumeClaim	DELETE /api/v1/namespaces/{namespace}/cloudpersistentvolumeclaims/{name}	cce:storage:delete	√	√

**Tabla 16-7 Addon**

Permiso	API	Acción	Proyectos de IAM	Proyecto empresarial
Creación de una instancia de complemento	POST /api/v3/addons	cce:addonInstance:create	√	√
Obtención de información sobre una instancia de complemento	GET /api/v3/addons/{id}?cluster_id={cluster_id}	cce:addonInstance:get	√	√
Listado de todas las instancias de complementos	GET /api/v3/addons?cluster_id={cluster_id}	cce:addonInstance:list	√	√
Eliminación de una instancia de complemento	DELETE /api/v3/addons/{id}?cluster_id={cluster_id}	cce:addonInstance:delete	√	√
Actualización de una instancia de complemento	PUT /api/v3/addons/{id}	cce:addonInstance:update	√	√

**Tabla 16-8** Quota

Permiso	API	Acción	Proyectos de IAM	Proyecto empresarial
Consulta de detalles de cuota	GET /api/v3/projects/{project_id}/quotas	cce:quota:get	√	√

## 16.2 Permisos de clúster (basados en IAM)

Los permisos a nivel de clúster de CCE se asignan según las **IAM system policies** y **custom policies**. Puede utilizar grupos de usuarios para asignar permisos a los usuarios de IAM.

---

 **ATENCIÓN**

**Cluster permissions** se configuran solo para recursos relacionados con clústeres (como clústeres y nodos). También debe configurar los **permisos de espacio de nombres** para operar los recursos de Kubernetes (como cargas de trabajo y Services).

---

### Requisitos previos

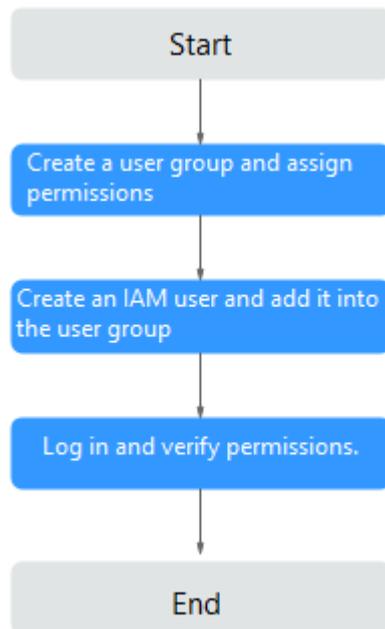
- Antes de conceder permisos a grupos de usuarios, familiarícese con las políticas del sistema que aparecen en **Gestión de permisos**. Para ver las políticas del sistema de otros servicios, consulte **Permisos del sistema**.
- Un usuario con el rol de Security Administrator (por ejemplo, su cuenta) tiene todos los permisos de IAM excepto el cambio de rol. Solo estos usuarios pueden ver los grupos de usuarios y sus permisos en la página **Permissions** de la consola de CCE.

### Configuración

En la consola de CCE, cuando elija **Permissions > Cluster-Level Permissions** para crear un grupo de usuarios, se le dirigirá a la consola de IAM para completar el proceso. Después de crear el grupo de usuarios y configurar sus permisos, puede ver la información en la página de ficha **Cluster-Level Permissions**. Esta sección describe las operaciones en IAM.

## Flujo del proceso

Figura 16-2 Proceso de asignación de permisos de CCE



1. **Cree un grupo de usuarios y asigne los permisos.**

Cree un grupo de usuarios en la consola de IAM y asigne los permisos de CCE, por ejemplo, la política **CCEReadOnlyAccess** al grupo.

**NOTA**

CCE se despliega por región. En la consola de IAM, seleccione **Region-specific projects** al asignar permisos de CCE.

2. **Cree un usuario y agréguelo a un grupo de usuarios.**

Cree un usuario en la consola de IAM y agregue el usuario al grupo creado en **1**.

3. **Inicie sesión** y verifique los permisos.

Inicie sesión en la consola de gestión como el usuario que creó y compruebe que el usuario tiene los permisos asignados.

- Inicie sesión en la consola de gestión, cambie a la consola de CCE y compre un clúster. Si no lo hace (suponiendo que solo se asigna el permiso **CCEReadOnlyAccess**), la política **CCEReadOnlyAccess** entra en vigor.
- Cambie a la consola de cualquier otro servicio. Si aparece un mensaje que indica que no tiene los permisos necesarios para acceder al servicio, la política **CCEReadOnlyAccess** entra en vigor.

## Roles definidos por el sistema

Los roles son un tipo de mecanismo de autorización de grano grueso que define permisos de nivel de servicio en función de las responsabilidades del usuario. Solo un número limitado de roles de nivel de servicio están disponibles para autorización. Los roles no son ideales para la autorización detallada y el acceso con privilegios mínimos.

El rol de sistema preestablecido para CCE en IAM es **CCEAdministrator**. Al asignar este rol a un grupo de usuarios, también debe seleccionar otros roles y políticas de los que depende este rol, como **Tenant Guest**, **Server Administrator**, **ELB Administrator**, **OBS Administrator**, **SFS Administrator**, **SWR Admin** y **APM FullAccess**. Para obtener más información acerca de las dependencias, vea [Permisos del sistema](#).

## Políticas definidas por el sistema

Las políticas de sistema preestablecidas para CCE en IAM son **CCEFullAccess** y **CCEReadOnlyAccess**.

- **CCE FullAccess**: permisos de operación comunes en los recursos de clúster de CCE, excluidos los permisos de espacio de nombres para los clústeres (con Kubernetes RBAC habilitado) y las operaciones de administrador con privilegios, como la configuración de delegación y la generación de certificados de clúster
- **CCE ReadOnlyAccess**: permisos para ver los recursos del clúster de CCE, excluidos los permisos a nivel de espacio de nombres de los clústeres (con Kubernetes RBAC habilitado)

**Tabla 16-9** Permisos asignados por CCEFullAccess

Acción	Acción específica	Descripción
cce:*:*	cce:cluster:create	Crear un clúster.
	cce:cluster:delete	Eliminar un clúster.
	cce:cluster:update	Actualizar un clúster, por ejemplo, actualizar los parámetros de programación de nodos de clúster y proporcionar soporte RBAC a los clústeres.
	cce:cluster:upgrade	Actualizar un clúster.
	cce:cluster:start	Despertar un grupo.
	cce:cluster:stop	Hibernar un clúster.
	cce:cluster:list	Enumerar todos los clústeres.
	cce:cluster:get	Consultar detalles del clúster.
	cce:node:create	Agregar un nodo de clúster.
	cce:node:delete	Eliminar uno o más nodos.
	cce:node:update	Actualizar un nodo de clúster, por ejemplo, actualizar el nombre del nodo.
	cce:node:get	Consultar detalles del nodo.
	cce:node:list	Listar todos los nodos.
	cce:nodepool:create	Crear un grupo de nodos.
cce:nodepool:delete	Eliminar un grupo de nodos.	

Acción	Acción específica	Descripción
	cce:nodepool:update	Actualizar información acerca de un grupo de nodos.
	cce:nodepool:get	Obtener información acerca de un grupo de nodos.
	cce:nodepool:list	Enumerar todos los grupos de nodos de un clúster.
	cce:release:create	Crear una versión.
	cce:release:delete	Eliminar una versión.
	cce:release:update	Actualizar una versión.
	cce:job:list	Enumerar todos los trabajos de clúster.
	cce:job:delete	Eliminar uno o más trabajos de clúster.
	cce:job:get	Consultar un trabajo de clúster especificado.
	cce:storage:create	Crear un volumen de almacenamiento.
	cce:storage:delete	Eliminar un volumen de almacenamiento.
	cce:storage:list	Listar todos los volúmenes.
	cce:addonInstance:create	Crear una instancia de complemento.
	cce:addonInstance:delete	Eliminar una instancia de complemento.
	cce:addonInstance:update	Actualizar una instancia de complemento.
	cce:addonInstance:get	Consultar detalles de la instancia del complemento.
	cce:addonTemplate:get	Consultar detalles de plantilla adicional.
	cce:addonInstance:list	Enumerar todas las instancias de complemento.
	cce:addonTemplate:list	Enumerar todas las plantillas adicionales.
	cce:chart:list	Listar todos los gráficos.
	cce:chart:delete	Eliminar un gráfico.
	cce:chart:update	Actualizar un gráfico.
	cce:chart:upload	Subir un gráfico.
	cce:chart:get	Obtener información sobre un gráfico.

Acción	Acción específica	Descripción
	cce:release:get	Obtener información acerca de un comunicado de prensa.
	cce:release:list	Listar todos los lanzamientos.
	cce:userAuthorization:get	Obtener la autorización del usuario de CCE.
	cce:userAuthorization:create	Crear la autorización de usuario de CCE.
ecs:*:*	-	Realizar todas las operaciones en Elastic Cloud Server (ECS).
evs:*:*	-	Realizar todas las operaciones en Elastic Volume Service (EVS). Los discos de EVS se pueden conectar a servidores en la nube y ampliarlos a una mayor capacidad cuando sea necesario.
vpc:*:*	-	Realizar todas las operaciones en Virtual Private Cloud (VPC), incluidos los ELB. Un clúster debe ejecutarse en una VPC. Al crear un espacio de nombres, debe crear o asociar una VPC para el espacio de nombres de modo que todos los contenedores del espacio de nombres se ejecuten en la VPC.
sfs:*:get*	-	Consultar los detalles de recursos del Scalable File Service (SFS).
sfs:shares:Share Action	-	Compartir recursos de SFS para escalar.
aom:*:get	-	Ver detalles de recursos de Application Operations Management (AOM).
aom:*:list	-	Enumerar los recursos de AOM.
aom:autoScalingRule:*	-	Realizar todas las operaciones en las reglas de ajuste automático de AOM.
apm:icmgr:*	-	Realizar operaciones en el ICAgent en Application Performance Management (APM).
lts:*:*	-	Realizar todas las operaciones en Log Tank Service (LTS).

**Tabla 16-10** Permisos asignados por CCEReadOnlyAccess

Acción	Acción específica	Descripción
cce:*.get	cce:cluster:get	Consultar detalles del clúster.
	cce:node:get	Consultar detalles del nodo.
	cce:job:get	Consultar un trabajo de clúster especificado.
	cce:addonInstance:get	Consultar detalles de la instancia del complemento.
	cce:addonTemplate:get	Consultar detalles de plantilla adicional.
	cce:chart:get	Obtener información sobre un gráfico.
	cce:nodepool:get	Obtener información acerca de un grupo de nodos.
	cce:release:get	Obtener información acerca de un comunicado de prensa.
	cce:userAuthorization:get	Obtener la autorización del usuario de CCE.
cce:*.list	cce:cluster:list	Enumerar todos los clústeres.
	cce:node:list	Listar todos los nodos.
	cce:job:list	Enumerar todos los trabajos de clúster.
	cce:addonInstance:list	Enumerar todas las instancias de complemento.
	cce:addonTemplate:list	Enumerar todas las plantillas adicionales.
	cce:chart:list	Listar todos los gráficos.
	cce:nodepool:list	Enumerar todos los grupos de nodos de un clúster.
	cce:release:list	Listar todos los lanzamientos.
	cce:storage:list	Listar todos los volúmenes.
cce:kubernetes:*	-	Realizar las operaciones en todos los recursos de Kubernetes. Para obtener más información, consulte <a href="#">Permisos de espacio de nombres</a> .
ecs:*.get	-	Ver detalles sobre todos los recursos de ECS. Un ECS con varios discos de EVS es un nodo de clúster en CCE.
ecs:*.list	-	Enumerar todos los recursos de ECS.
bms:*.get*	-	Ver detalles sobre todos los recursos de BMS.
bms:*.list	-	Listar todos los recursos de BMS.

Acción	Acción específica	Descripción
evs:*.get	-	Ver detalles de todos los recursos de disco de EVS. Los discos de EVS se pueden conectar a servidores en la nube y ampliarlos a una mayor capacidad cuando sea necesario.
evs:*.list	-	Listar todos los recursos de EVS.
evs:*.count	-	-
vpc:*.get	-	Ver detalles de todos los recursos de VPC (incluidos los ELB). Un clúster debe ejecutarse en una VPC. Al crear un espacio de nombres, debe crear o asociar una VPC para el espacio de nombres de modo que todos los contenedores del espacio de nombres se ejecuten en la VPC.
vpc:*.list	-	Enumerar todos los recursos de VPC (incluidos los ELB).
sfs:*.get*	-	Ver detalles de recursos de SFS.
sfs:shares:Share Action	-	Compartir recursos de SFS para escalar.
aom:*.get	-	Ver detalles de recursos de AOM.
aom:*.list	-	Listar todos los recursos de AOM.
aom:autoScalingRule:*	-	Realizar todas las operaciones en las reglas de ajuste automático de AOM.
lts:*.get	-	Ver detalles sobre todos los recursos de LTS.
lts:*.list	-	Listar todos los recursos de LTS.

## Políticas personalizadas

Las políticas personalizadas se pueden crear como un suplemento a las políticas definidas por el sistema de CCE. Para ver las acciones que se pueden agregar a las políticas personalizadas, consulte [Políticas de permisos y acciones admitidas](#).

Puede crear las políticas personalizadas de cualquiera de las siguientes maneras:

- Visual editor: Seleccione servicios en la nube, acciones, recursos y condiciones de solicitud. Esto no requiere conocimiento de la sintaxis de políticas.
- JSON: Edite las políticas de JSON desde cero o basándose en una política existente.

Para obtener más información, consulte [Creación de una política personalizada](#). Esta sección proporciona ejemplos de políticas de CCE personalizadas comunes.

### Ejemplo de políticas personalizadas:

- **Ejemplo 1: Creación de un clúster denominado test**

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cce:cluster:create"
      ]
    }
  ]
}
```

- **Ejemplo 2: Detención de la eliminación de nodos**

Una política con solo los permisos "Deny" debe usarse junto con otras políticas para que surtan efecto. Si los permisos asignados a un usuario contienen tanto "Allow" como "Deny", los permisos "Deny" tienen prioridad sobre los permisos "Allow".

El siguiente método se puede utilizar si necesita asignar permisos de la política **CCEFullAccess** a un usuario pero desea evitar que el usuario elimine nodos (**cce:node:delete**). Cree una política personalizada para denegar la eliminación de nodos y adjunte ambas políticas al grupo al que pertenece el usuario. A continuación, el usuario puede realizar todas las operaciones en CCE excepto la eliminación de nodos. A continuación se muestra un ejemplo de una política de denegación:

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "cce:node:delete"
      ]
    }
  ]
}
```

- **Ejemplo 3: Definición de permisos para varios servicios en una política**

Una política personalizada puede contener las acciones de varios servicios que son de tipo global o de nivel de proyecto. A continuación se muestra una política de ejemplo que contiene acciones de varios servicios:

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Action": [
        "ecs:cloudServers:resize",
        "ecs:cloudServers:delete",
        "ecs:cloudServers:delete",
        "ims:images:list",
        "ims:serverImages:create"
      ],
      "Effect": "Allow"
    }
  ]
}
```

## Permisos de clúster de CCE y proyectos empresariales

CCE admite la gestión de recursos y la asignación de permisos por clúster y proyecto empresarial.

Tenga en cuenta que:

- Los proyectos de IAM se basan en el aislamiento físico de los recursos, mientras que los proyectos empresariales proporcionan los grupos lógicos globales de recursos que satisfacen mejor las necesidades reales de las empresas. Además, las políticas de IAM se pueden gestionar en función de proyectos empresariales. Por lo tanto, se recomienda utilizar proyectos de empresa para la gestión de permisos. Para obtener más información, consulte [Creación de un proyecto de empresa](#).
- Cuando hay proyectos de IAM y proyectos empresariales, IAM coincide preferentemente con las políticas de proyecto de IAM.
- Al crear un clúster o nodo con recursos de nube comprados, asegúrese de que los usuarios de IAM tienen los permisos necesarios en el proyecto empresarial para usar estos recursos. De lo contrario, es posible que no se cree el clúster o el nodo.
- Si un recurso no admite proyectos de empresa, los permisos concedidos al recurso no tendrán efecto.

Tipo de recurso	Nombre del recurso	Descripción
Apoyo a proyectos empresariales	cluster	Clúster
	node	Nodo
	nodepool	Grupo de nodos
	job	Trabajo
	tag	Etiqueta de clúster
	addonInstance	Instancia de complementos
	release	Versión de Helm
	storage	Almacenamiento
No admite proyectos empresariales	quota	Cuota de clúster
	chart	Gráfico
	addonTemplate	Plantilla de complemento

## Permisos de clúster de CCE y IAM RBAC

CCE es compatible con los roles del sistema de IAM para la gestión de permisos. Se recomienda utilizar las políticas detalladas proporcionadas por IAM para simplificar la gestión de permisos.

CCE admite las siguientes funciones:

- Funciones básicas de IAM:
  - te\_admin (Tenant Administrator): Los usuarios con este rol pueden invocar a todas las API de todos los servicios excepto IAM.
  - readonly (Tenant Guest): los usuarios con este rol pueden invocar a las API con los permisos de solo lectura de todos los servicios excepto IAM.
- Rol de administrador de CCE personalizado: CCE Administrator
- Las API de CCE están diseñadas para ser compatibles con tres roles de sistema heredados de [ServiceStage](#) (SvcStg Administrator, SvcStg Developer y SvcStg

Operator). Actualmente, CCE y ServiceStage se han adaptado completamente a las políticas detalladas de IAM para la gestión de permisos. Por lo tanto, no se recomienda utilizar estas funciones heredadas para la gestión de permisos. Estas funciones de ServiceStage se describen de la siguiente manera:

- SvcStg Administrator: este rol tiene los mismos permisos que el rol CCE Administrator, excepto que los usuarios con este rol no tienen los permisos de nivel de espacio de nombres de forma predeterminada (Kubernetes RBAC).
- SvcStg Developer: Este rol tiene los mismos permisos que el rol CCE Administrator, excepto que el usuario con este rol no tiene el permiso de nivel de espacio de nombres de forma predeterminada (Kubernetes RBAC).
- SvcStg Operator: este rol tiene los permisos de solo lectura en CCE, excluidos los permisos de nivel de espacio de nombres de los clústeres de forma predeterminada.

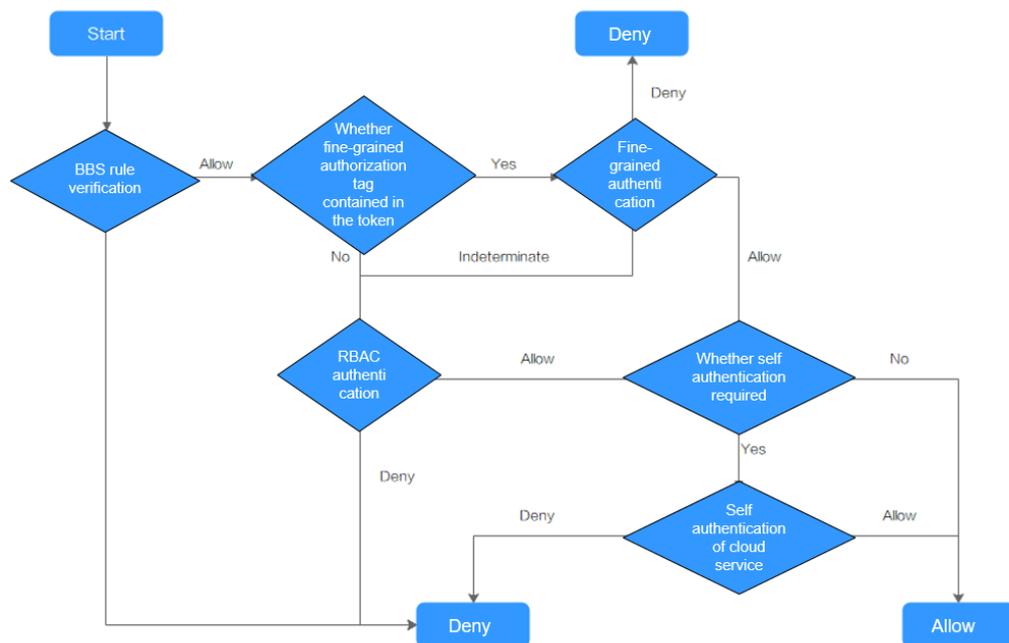
**NOTA**

- Tenant Administrator y Tenant Guest son los roles especiales del sistema de IAM. Después de configurar cualquier sistema o política personalizada, Tenant Administrator y Tenant Guest entran en vigor como políticas del sistema para lograr la compatibilidad con escenarios IAM RBAC y ABAC.
- Si un usuario tiene el rol de sistema de Tenant Administrator o de CCE Administrator, el usuario tiene los permisos de cluster-admin en Kubernetes RBAC y los permisos no se pueden quitar después de crear el clúster.

Si el usuario es el creador del clúster, los permisos de cluster-admin en Kubernetes RBAC se otorgan al usuario de forma predeterminada. Los permisos se pueden quitar manualmente después de crear el clúster.

- Método 1: Elija **Permissions Management > Namespace-Level Permissions > Delete** con el mismo rol que el creador de clústeres en la consola de CCE.
- Método 2: Elimine **ClusterRoleBinding: cluster-creator** con la API o kubectl.

Cuando las políticas RBAC e IAM coexisten, la lógica de autenticación de back-end para las API abiertas o las operaciones de consola en CCE es la siguiente:



**⚠ ATENCIÓN**

Ciertas API de CCE implican permisos a nivel de espacio de nombres u operaciones de clave y, por lo tanto, requieren los permisos especiales:

Uso de clusterCert para obtener el clúster de kubeconfig: cceadm/teadmin

## 16.3 Permisos de espacio de nombres (basados en Kubernetes RBAC)

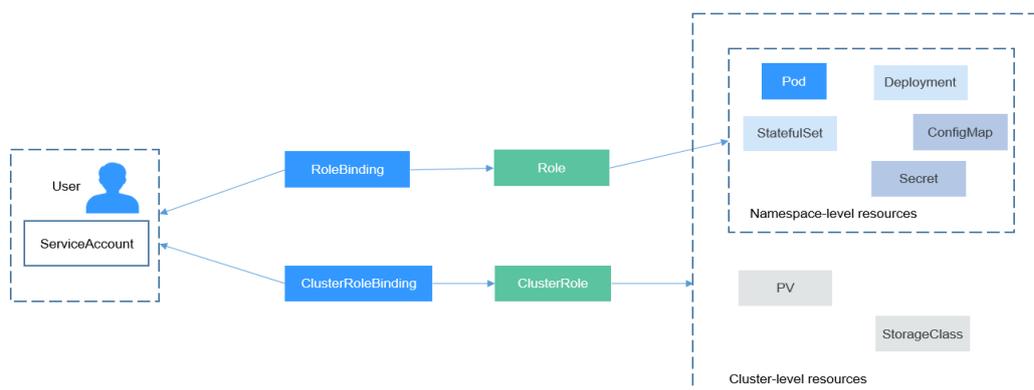
### Permisos de espacio de nombres (basados en Kubernetes RBAC)

Puede regular el acceso de los usuarios o los grupos de usuarios a los recursos de Kubernetes en un único espacio de nombres basado en sus roles de Kubernetes RBAC. La API de RBAC declara cuatro tipos de objetos de Kubernetes: Role, ClusterRole, RoleBinding y ClusterRoleBinding, que se describen a continuación:

- **Función:** define un conjunto de reglas para acceder a los recursos de Kubernetes en un espacio de nombres.
- **RoleBinding:** define la relación entre usuarios y roles.
- **ClusterRole:** define un conjunto de reglas para acceder a los recursos de Kubernetes en un clúster (incluidos todos los espacios de nombres).
- **ClusterRoleBinding:** define la relación entre los usuarios y los roles de clúster.

Role y ClusterRole especifican acciones que se pueden realizar en los recursos específicos. RoleBinding y ClusterRoleBinding vinculan roles a usuarios específicos, grupos de usuarios o ServiceAccounts. Ilustración:

**Figura 16-3** Vinculación de roles



En la consola de CCE, puede asignar permisos a un usuario o grupo de usuarios para tener acceso a recursos en uno o varios espacios de nombres. De forma predeterminada, la consola de CCE proporciona el ClusterRoles siguiente:

- **view (solo lectura):** permiso de solo lectura en la mayoría de los recursos de todos los espacios de nombres o seleccionados.

- edit (desarrollo): permisos de lectura y escritura en la mayoría de los recursos en todos o en espacios de nombres seleccionados. Si este ClusterRole está configurado para todos los espacios de nombres, su capacidad es la misma que el permiso de O&M.
- admin (O&M): permisos de lectura y escritura en la mayoría de los recursos de todos los espacios de nombres, y permisos de solo lectura en nodos, volúmenes de almacenamiento, espacios de nombres y gestión de cuotas.
- cluster-admin (administrador): permisos de lectura y escritura en todos los recursos de todos los espacios de nombres.

## Cluster Permissions (basados en IAM) y Namespace Permissions (basados en Kubernetes RBAC)

Los usuarios con diferentes permisos de clúster (asignados mediante IAM) tienen diferentes permisos de espacio de nombres (asignados mediante Kubernetes RBAC). [Tabla 16-11](#) enumera los permisos de espacio de nombres de diferentes usuarios.

**Tabla 16-11** Diferencias en los permisos de espacio de nombres

Usuario	Clústeres de v1.13 y posteriores
Usuario con los permisos de Tenant Administrator (por ejemplo, una cuenta)	Todos los permisos de espacio de nombres
Usuario de IAM con el rol CCE Administrator	Todos los permisos de espacio de nombres
Usuario de IAM con el rol CCE FullAccess o CCE ReadOnlyAccess	Requiere la autorización RBAC de Kubernetes.
Usuario de IAM con el rol Tenant Guest	Requiere la autorización RBAC de Kubernetes.

## Precauciones

- La autorización de RBAC de Kubernetes se puede usar para clústeres de v1.11.7-r2 y posteriores. Asegúrese de haber desplegado una versión de clúster compatible. Para obtener más información sobre la actualización de un clúster, consulte [Actualización de sustitución/rodamiento \(versión 1.13\)](#).
- Después de crear un clúster de v1.11.7-r2 o posterior, CCE le asigna automáticamente el permiso de cluster-admin, lo que significa que tiene control total sobre todos los recursos de todos los espacios de nombres del clúster. El ID de un usuario federado cambia en cada inicio de sesión y cierre de sesión. Por lo tanto, el usuario con los permisos se muestra como eliminado. En este caso, no elimine los permisos. De lo contrario, la autenticación falla. Se recomienda conceder el permiso de administrador del clúster a un grupo de usuarios en CCE y agregar usuarios federados al grupo de usuarios.
- Un usuario con el rol Security Administrator tiene todos los permisos de IAM excepto el cambio de rol. Por ejemplo, una cuenta del grupo de usuarios admin tiene esta función de forma predeterminada. Solo estos usuarios pueden asignar permisos en la página **Permissions** de la consola de CCE.

## Configuración de permisos de espacio de nombres (en la consola)

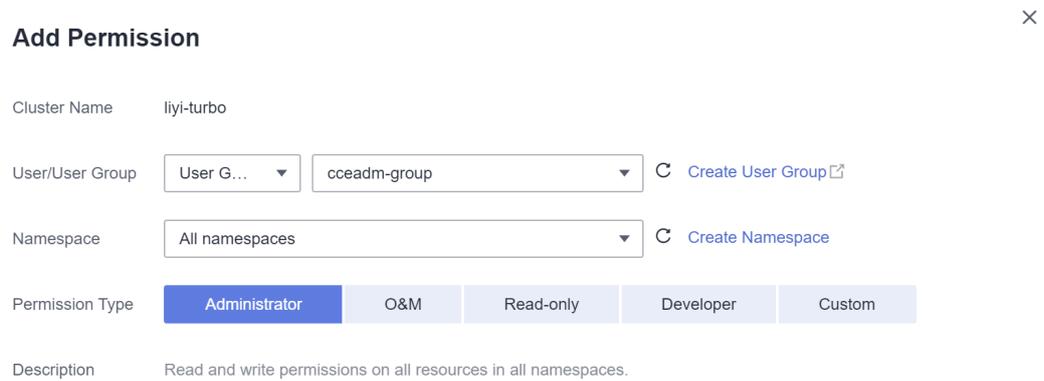
Puede regular el acceso de los usuarios o los grupos de usuarios a los recursos de Kubernetes en un único espacio de nombres basado en sus roles de Kubernetes RBAC.

- Paso 1** Inicie sesión en la consola de CCE. En el panel de navegación, elija **Permissions**.
- Paso 2** Seleccione un clúster para el que desee agregar permisos en la lista desplegable de la derecha.
- Paso 3** Haga clic en **Add Permissions** en la esquina superior derecha.
- Paso 4** Confirme el nombre del clúster y seleccione el espacio de nombres para el que asignar permisos. Por ejemplo, seleccione **All namespaces**, el usuario o grupo de usuarios de destino y seleccione los permisos.

### **NOTA**

Si no tiene permisos de IAM, no puede seleccionar usuarios o grupos de usuarios al configurar permisos para otros usuarios o grupos de usuarios. En este caso, puede introducir un ID de usuario o un ID de grupo de usuarios.

**Figura 16-4** Configuración de permisos de espacio de nombres



**Add Permission** ×

Cluster Name

User/User Group   [Create User Group](#)

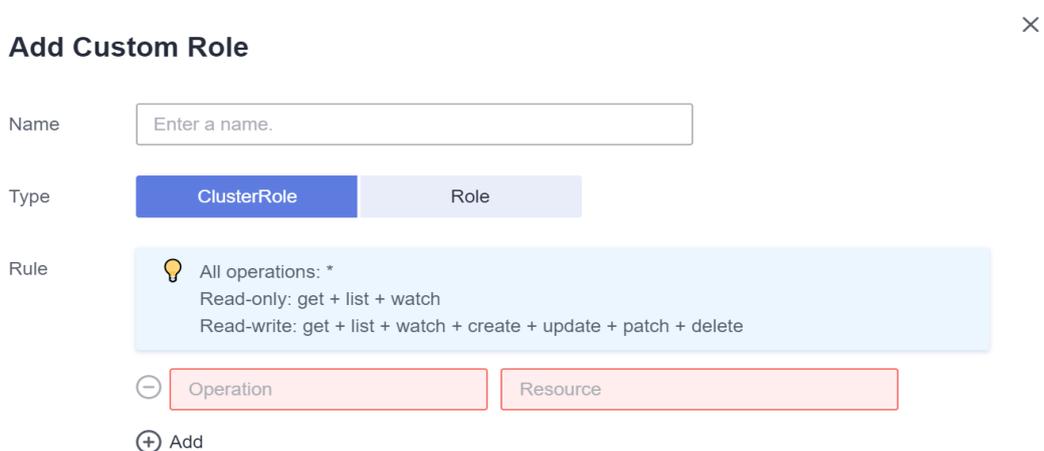
Namespace  [Create Namespace](#)

Permission Type **Administrator** O&M Read-only Developer Custom

Description

Los permisos se pueden personalizar según sea necesario. Después de seleccionar **Custom** para **Permission Type**, haga clic en **Add Custom Role** a la derecha del parámetro **Custom**. En el cuadro de diálogo que se muestra, escriba un nombre y seleccione una regla. Una vez creada la regla personalizada, puede seleccionar un valor en el cuadro de lista desplegable **Custom**.

**Figura 16-5** Permiso personalizado



**Add Custom Role** ×

Name

Type **ClusterRole** Role

Rule  All operations: \*  
Read-only: get + list + watch  
Read-write: get + list + watch + create + update + patch + delete

**Paso 5** Haga clic en **OK**.

----Fin

## Uso de kubectl para configurar permisos de espacio de nombres

### NOTA

Cuando se accede a un clúster mediante kubectl, CCE utiliza el archivo kubeconfig.json generado en el clúster para la autenticación. Este archivo contiene información del usuario, basada en la cual CCE determina qué recursos de Kubernetes puede acceder kubectl. Los permisos registrados en un archivo kubeconfig.json varían de usuario a usuario. [Cluster Permissions \(basados en IAM\) y Namespace Permissions \(basados en Kubernetes RBAC\)](#) muestra los permisos que tiene un usuario.

Además de cluster-admin, admin, edit, y view, puede definir Roles y RoleBindings para configurar los permisos para agregar, eliminar, modificar y consultar recursos, como pods, implementaciones y servicios, en el espacio de nombres.

El procedimiento para crear un Role es muy simple. Para ser específico, especifique un espacio de nombres y, a continuación, defina las reglas. Las reglas del siguiente ejemplo son permitir las operaciones GET y LIST en los pods en el espacio de nombres predeterminado.

```
kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  namespace: default # Namespace
  name: role-example
rules:
- apiGroups: [""]
  resources: ["pods"] # The pod can be accessed.
  verbs: ["get", "list"] # The GET and LIST operations can
  be performed.
```

- **apiGroups** indica el grupo de API al que pertenece el recurso.
- **resources** indica los recursos que se pueden operar. Se admiten pods, Deployments, ConfigMaps y otros recursos de Kubernetes.
- **verbs** indica las operaciones que se pueden realizar. **get** indica la consulta de un objeto específico, y **list** indica la lista de todos los objetos de un tipo determinado. Otras opciones de valor incluyen **create**, **update** y **delete**.

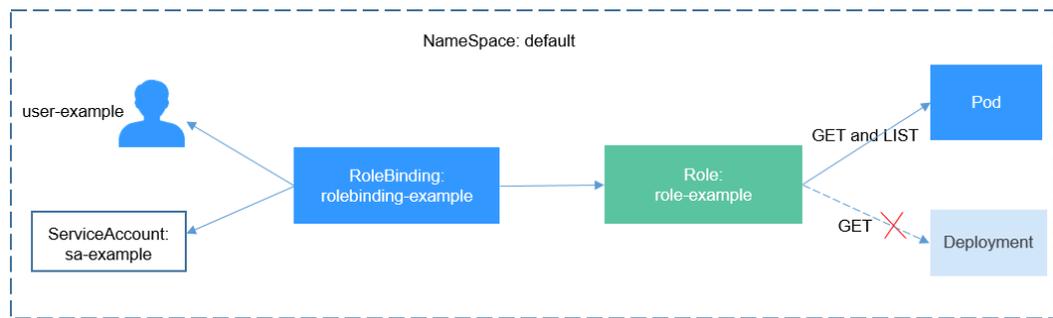
Para obtener más información, consulte [Uso de la autorización RBAC](#).

Después de crear un Role, puede enlazar el Role a un usuario específico, que se denomina RoleBinding. Lo siguiente es un ejemplo.

```
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: RoleBinding-example
  namespace: default
  annotations:
    CCE.com/IAM: 'true'
roleRef:
  kind: Role
  name: role-example
  apiGroup: rbac.authorization.k8s.io
subjects:
- kind: User
  name: 0c97ac3cb280f4d91fa7c0096739e1f8 # User ID of the user-example
  apiGroup: rbac.authorization.k8s.io
```

La sección **subjects** vincula un rol con un usuario de IAM para que el usuario de IAM pueda obtener los permisos definidos en el rol, como se muestra en la siguiente figura.

**Figura 16-6** Un RoleBinding vincula el rol al usuario.



También puede especificar un grupo de usuarios en la sección **subjects**. En este caso, todos los usuarios del grupo de usuarios obtienen los permisos definidos en el rol.

```
subjects:
- kind: Group
  name: 0c96fad22880f32a3f84c009862af6f7 # User group ID
  apiGroup: rbac.authorization.k8s.io
```

Utilice el ejemplo de usuario de IAM para conectarse al clúster y obtener la información del pod. A continuación se muestra un ejemplo de la información de pod devuelta.

```
# kubectl get pod
NAME                                READY   STATUS    RESTARTS   AGE
deployment-389584-2-6f6bd4c574-2n9rk  1/1     Running   0           4d7h
deployment-389584-2-6f6bd4c574-7s5qw  1/1     Running   0           4d7h
deployment-3895841-746b97b455-86g77   1/1     Running   0           4d7h
deployment-3895841-746b97b455-twvnp   1/1     Running   0           4d7h
nginx-658dff48ff-7rkph                 1/1     Running   0           4d9h
nginx-658dff48ff-njdhj                 1/1     Running   0           4d9h
# kubectl get pod nginx-658dff48ff-7rkph
NAME                                READY   STATUS    RESTARTS   AGE
nginx-658dff48ff-7rkph              1/1     Running   0           4d9h
```

Intente consultar Deployments y Services en el espacio de nombres. El resultado muestra que **user-example** no tiene los permisos necesarios. Intente consultar los pods en el espacio de nombres kube-system. El resultado muestra que **user-example** tampoco tiene los permisos necesarios. Esto indica que el usuario de IAM **user-example** solo tiene los permisos GET y LIST Pod en el espacio de nombres predeterminado, que es el mismo que se esperaba.

```
# kubectl get deploy
Error from server (Forbidden): deployments.apps is forbidden: User
"0c97ac3cb280f4d91fa7c0096739e1f8" cannot list resource "deployments" in API
group "apps" in the namespace "default"
# kubectl get svc
Error from server (Forbidden): services is forbidden: User
"0c97ac3cb280f4d91fa7c0096739e1f8" cannot list resource "services" in API group
"" in the namespace "default"
# kubectl get pod --namespace=kube-system
Error from server (Forbidden): pods is forbidden: User
"0c97ac3cb280f4d91fa7c0096739e1f8" cannot list resource "pods" in API group "" in
the namespace "kube-system"
```

### Ejemplo: Asignación de todos los permisos de clúster (cluster-admin)

Puede usar el rol cluster-admin para asignar todos los permisos de un clúster. Este rol contiene los permisos para los recursos del clúster (como PV y StorageClasses).

**Figura 16-7** Asignar todos los permisos de clúster (cluster-admin)

En el siguiente ejemplo de salida de kubectl, se ha creado un ClusterRoleBinding y enlaza el rol de cluster-admin al grupo de usuarios **cce-role-group**.

```
# kubectl get clusterrolebinding
NAME
ROLE          AGE
clusterrole_cluster-admin_group0c96fad22880f32a3f84c009862af6f7  ClusterRole/
cluster-admin          61s

# kubectl get clusterrolebinding clusterrole_cluster-
admin_group0c96fad22880f32a3f84c009862af6f7 -oyaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  annotations:
    CCE.com/IAM: "true"
  creationTimestamp: "2021-06-23T09:15:22Z"
  name: clusterrole_cluster-admin_group0c96fad22880f32a3f84c009862af6f7
  resourceVersion: "36659058"
  selfLink: /apis/rbac.authorization.k8s.io/v1/clusterrolebindings/
clusterrole_cluster-admin_group0c96fad22880f32a3f84c009862af6f7
  uid: d6cd43e9-b4ca-4b56-bc52-e36346fc1320
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: 0c96fad22880f32a3f84c009862af6f7
```

Conéctese al clúster como usuario autorizado. Si se pueden consultar los PV y los StorageClasses, la configuración de permisos tiene efecto.

```
# kubectl get pv
No resources found
# kubectl get sc
NAME          PROVISIONER          RECLAIMPOLICY
VOLUMEBINDINGMODE  ALLOWVOLUMEEXPANSION  AGE
csi-disk          everest-csi-provisioner  Delete
Immediate        true                  75d
csi-disk-topology  everest-csi-provisioner  Delete
WaitForFirstConsumer  true                  75d
csi-nas           everest-csi-provisioner  Delete
Immediate        true                  75d
csi-obs           everest-csi-provisioner  Delete
Immediate        false                 75d
csi-sfsturbo      everest-csi-provisioner  Delete
Immediate        true                  75d
```

### Ejemplo: Asignación de todos los permisos de espacio de nombres (admin)

**admin** tiene todos los permisos en los espacios de nombres. Puede otorgar este rol a un usuario o grupo de usuarios para gestionar uno o todos los espacios de nombres.

**Figura 16-8** Asignar todos los permisos de espacio de nombres (admin)

En el siguiente ejemplo de salida de `kubectl`, se ha creado un `RoleBinding`, el rol de `admin` está enlazado al grupo de usuarios **`cce-role-group`** y el espacio de nombres de destino es el espacio de nombres predeterminado.

```
# kubectl get rolebinding
NAME                                                    ROLE                AGE
clusterrole_admin_group0c96fad22880f32a3f84c009862af6f7  ClusterRole/admin  18s
# kubectl get rolebinding clusterrole_admin_group0c96fad22880f32a3f84c009862af6f7
-yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  annotations:
    CCE.com/IAM: "true"
  creationTimestamp: "2021-06-24T01:30:08Z"
  name: clusterrole_admin_group0c96fad22880f32a3f84c009862af6f7
  namespace: default
  resourceVersion: "36963685"
  selfLink: /apis/rbac.authorization.k8s.io/v1/namespaces/default/rolebindings/clusterrole_admin_group0c96fad22880f32a3f84c009862af6f7
  uid: 6c6f46a6-8584-47da-83f5-9eef1f7b75d6
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: admin
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: 0c96fad22880f32a3f84c009862af6f7
```

Conéctese a un clúster como usuario autorizado. En este ejemplo, puede crear y consultar recursos en el espacio de nombres predeterminado, pero no puede consultar recursos en el espacio de nombres del `kube-system` o en los recursos del clúster.

```
# kubectl get pod
NAME                READY   STATUS    RESTARTS   AGE
test-568d96f4f8-brdrp  1/1     Running   0           33m
test-568d96f4f8-cgjqp  1/1     Running   0           33m
# kubectl get pod -nkube-system
Error from server (Forbidden): pods is forbidden: User
"0c97ac3cb280f4d91fa7c0096739e1f8" cannot list resource "pods" in API group "" in
the namespace "kube-system"
# kubectl get pv
Error from server (Forbidden): persistentvolumes is forbidden: User
"0c97ac3cb280f4d91fa7c0096739e1f8" cannot list resource "persistentvolumes" in
API group "" at the cluster scope
```

### Ejemplo: Asignación de permisos de espacio de nombres de solo lectura (view)

El rol de `view` tiene los permisos de solo lectura en un espacio de nombres. Puede asignar permisos a los usuarios para ver uno o varios espacios de nombres.

**Figura 16-9** Asignación de permisos de espacio de nombres de solo lectura (view)

En el siguiente ejemplo de salida de kubectl, se ha creado un RoleBinding, el rol de view está enlazado al grupo de usuarios **cce-role-group** y el espacio de nombres de destino es el espacio de nombres predeterminado.

```
# kubectl get rolebinding
NAME                                     ROLE                AGE
clusterrole_view_group0c96fad22880f32a3f84c009862af6f7  ClusterRole/view   7s

# kubectl get rolebinding clusterrole_view_group0c96fad22880f32a3f84c009862af6f7 -o yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  annotations:
    CCE.com/IAM: "true"
  creationTimestamp: "2021-06-24T01:36:53Z"
  name: clusterrole_view_group0c96fad22880f32a3f84c009862af6f7
  namespace: default
  resourceVersion: "36965800"
  selfLink: /apis/rbac.authorization.k8s.io/v1/namespaces/default/rolebindings/clusterrole_view_group0c96fad22880f32a3f84c009862af6f7
  uid: b86e2507-e735-494c-be55-c41a0c4ef0dd
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: view
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: 0c96fad22880f32a3f84c009862af6f7
```

Connect to the cluster as an authorized user. En este ejemplo, puede consultar recursos en el espacio de nombres predeterminado, pero no puede crear recursos.

```
# kubectl get pod
NAME                READY   STATUS    RESTARTS   AGE
test-568d96f4f8-brdrp  1/1    Running   0           40m
test-568d96f4f8-cgjqp  1/1    Running   0           40m
# kubectl run -i --tty --image tutum/dnsutils dnsutils --restart=Never --rm /bin/sh
Error from server (Forbidden): pods is forbidden: User "0c97ac3cb280f4d91fa7c0096739e1f8" cannot create resource "pods" in API group "" in the namespace "default"
```

## Ejemplo: Asignación de permisos para un objeto de recurso específico de Kubernetes

Puede asignar permisos a un objeto de recurso de Kubernetes específico, como pod, Deployment y Service. Para obtener más información, véase [Uso de kubectl para configurar permisos de espacio de nombres](#).

## 16.4 Ejemplo: Diseño y configuración de permisos para usuarios en un departamento

### Descripción general

El modo de programación de tareas distribuidas convencional está siendo reemplazado por Kubernetes. CCE le permite desplegar, gestionar y escalar fácilmente las aplicaciones en contenedores en la nube al proporcionarle soporte para usar Kubernetes.

Para ayudar a los administradores empresariales a gestionar los permisos de recursos en clústeres, CCE proporciona políticas de permisos multidimensionales y medidas de gestión. Los permisos de CCE se describen de las siguientes maneras:

- **Cluster-level permissions:** permite que un grupo de usuarios realice operaciones en clústeres, nodos, grupos de nodos, gráficos y complementos. Estos permisos se asignan según las políticas del sistema de IAM.
- **Namespace-level permissions:** permite a un usuario o grupo de usuarios realizar operaciones en recursos de Kubernetes, como cargas de trabajo, redes, almacenamiento y espacios de nombres. Estos permisos se asignan en función de Kubernetes RBAC.

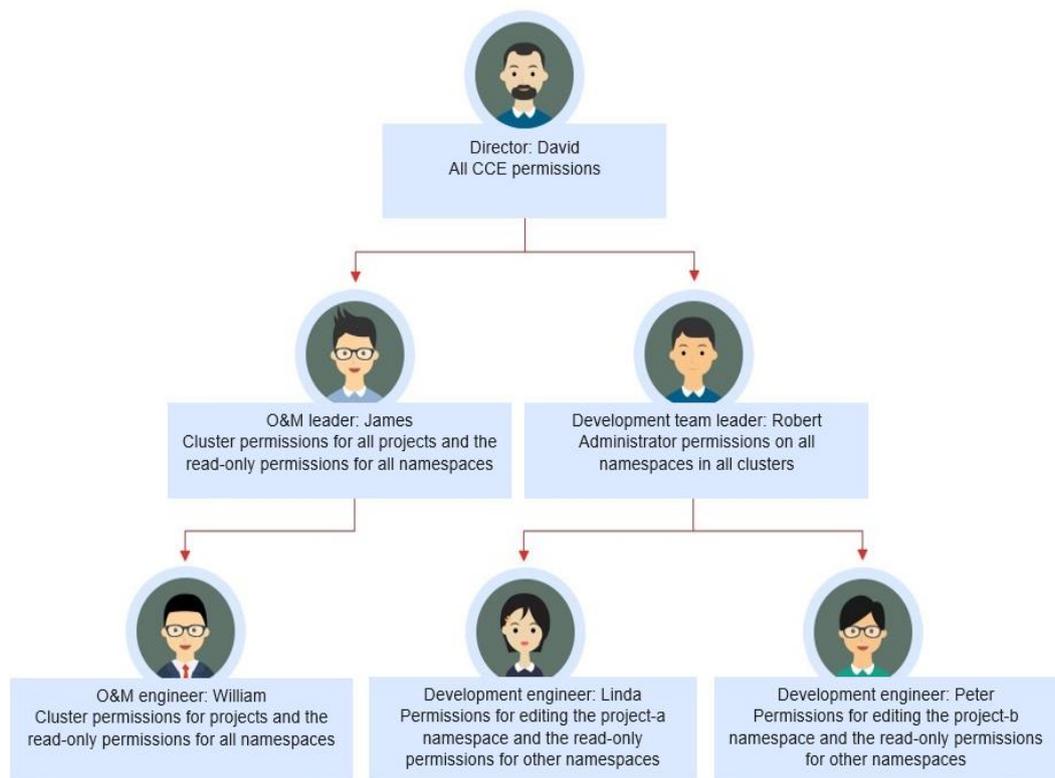
Los permisos de clúster y de espacio de nombres son independientes entre sí, pero deben usarse juntos. Los permisos establecidos para un grupo de usuarios se aplican a todos los usuarios del grupo de usuarios. Cuando se agregan varios permisos a un usuario o grupo de usuarios, tienen efecto al mismo tiempo (se usa el conjunto de unión).

### Diseño de permisos

A continuación se utiliza la compañía X como ejemplo.

Generalmente, una empresa tiene varios departamentos o proyectos, y cada departamento tiene varios miembros. Por lo tanto, debe diseñar cómo se asignarán los permisos a diferentes grupos y proyectos, y establecer un nombre de usuario para cada miembro para facilitar la configuración posterior de grupos de usuarios y permisos.

La siguiente figura muestra la estructura organizativa de un departamento de una empresa y los permisos que se asignarán a cada miembro:



## Director: David

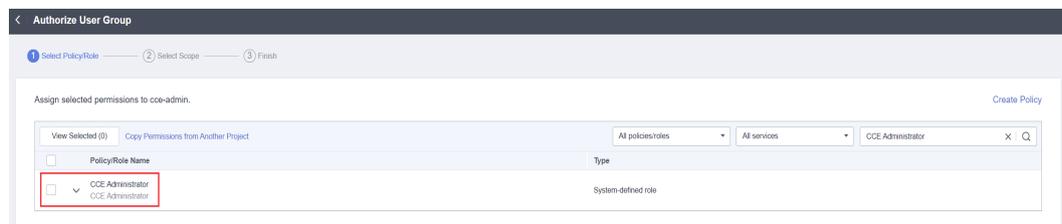
David es director de departamento de la compañía X. Para asignarle todos los permisos de CCE (tanto los permisos de clúster como de espacio de nombres), debe crear el grupo de usuarios **cce-admin** para David en la consola de IAM y asignar el rol de CCE Administrator.

### NOTA

**CCE Administrator:** Este rol tiene todos los permisos de CCE. No es necesario asignar otros permisos.

**CCE FullAccess and CCE ReadOnlyAccess:** Estas políticas están relacionadas con los permisos de gestión de clústeres y se configuran solo para recursos relacionados con clústeres (como clústeres y nodos). También debe configurar permisos de espacio de nombres para realizar operaciones en recursos de Kubernetes (como cargas de trabajo y Services).

**Figura 16-10** Asignar permisos al grupo de usuarios al que pertenece David

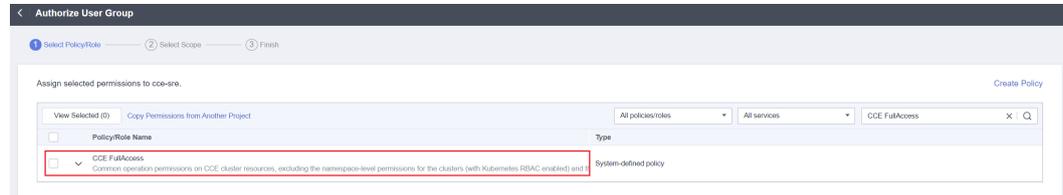


## Líder de O&M: James

James es líder del equipo O&M del departamento. Necesita los permisos de clúster para todos los proyectos y los permisos de solo lectura para todos los espacios de nombres.

Para asignar los permisos, cree un grupo de usuarios llamado **cce-sre** en la consola de IAM y agregue James a este grupo de usuarios. A continuación, asigne CCE FullAccess al grupo de usuarios **cce-sre** para que pueda realizar operaciones en clústeres en todos los proyectos.

**Figura 16-11** Asignar permisos al grupo de usuarios al que pertenece James



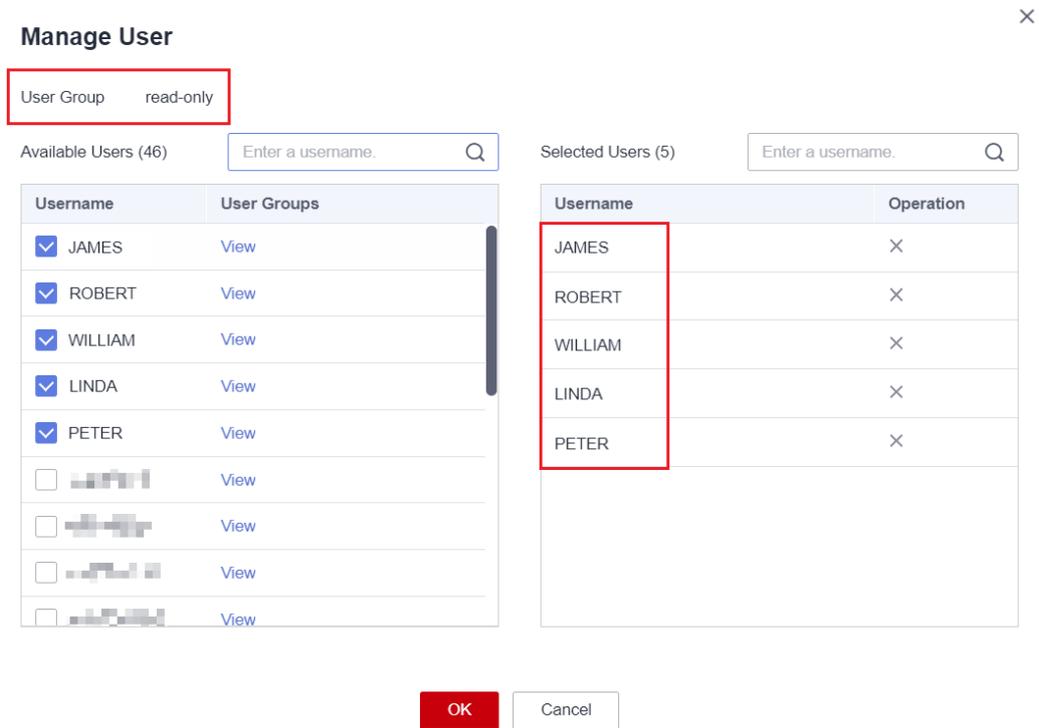
### Asignación de permisos de solo lectura en todos los clústeres y espacios de nombres a todos los líderes de equipo e ingenieros

Puede crear un grupo de usuarios de solo lectura denominado **read\_only** en la consola de IAM y agregar usuarios al grupo de usuarios.

- Aunque los ingenieros de desarrollo Linda y Peter no requieren permisos de gestión de clústeres, todavía necesitan ver los datos en la consola de CCE. Por lo tanto, se requiere el permiso de clúster de solo lectura.
- Para el ingeniero de O&M, William, asigne el permiso de solo lectura en clústeres en este paso.
- El líder del equipo O&M James ya tiene los permisos de gestión en todos los clústeres. Puede agregarlo al grupo de usuarios **read\_only** para asignarle el permiso de solo lectura en clústeres.

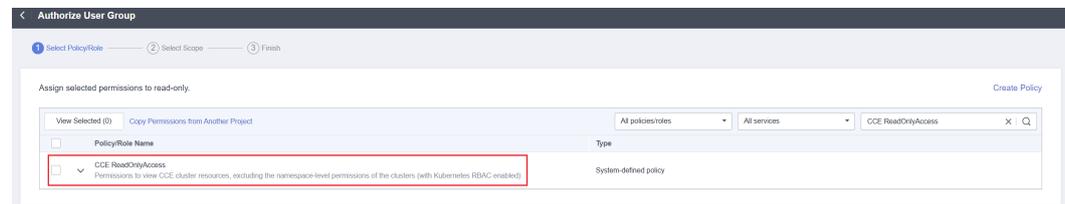
Los usuarios James, Robert, William, Linda y Peter se agregan al grupo de usuarios **read\_only**.

**Figura 16-12** Agregar usuarios al grupo de usuarios de solo lectura



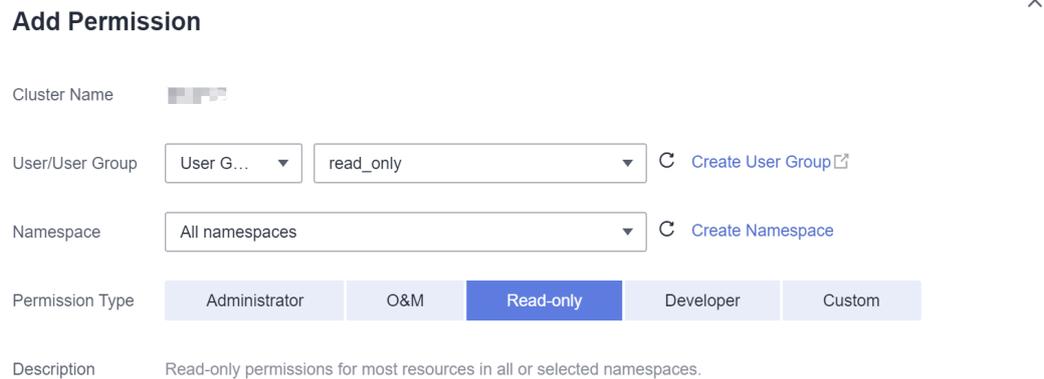
Asigne el permiso de solo lectura en clústeres al **read\_only** del grupo de usuarios.

**Figura 16-13** Asignar el permiso de solo lectura en clústeres al grupo de usuarios



Vuelva a la consola de CCE y agregue el permiso de solo lectura en espacios de nombres al grupo de usuarios **read\_only** al que pertenecen los cinco usuarios. Elija **Permissions** en la consola de CCE y asigne la política de solo lectura al grupo de usuarios **read\_only** para cada clúster.

**Figura 16-14** Asignar el permiso de solo lectura en espacios de nombres al grupo de usuarios



**Add Permission** ×

Cluster Name [REDACTED]

User/User Group User G... ▼ read\_only ▼ C Create User Group

Namespace All namespaces ▼ C Create Namespace

Permission Type Administrator O&M Read-only Developer Custom

Description Read-only permissions for most resources in all or selected namespaces.

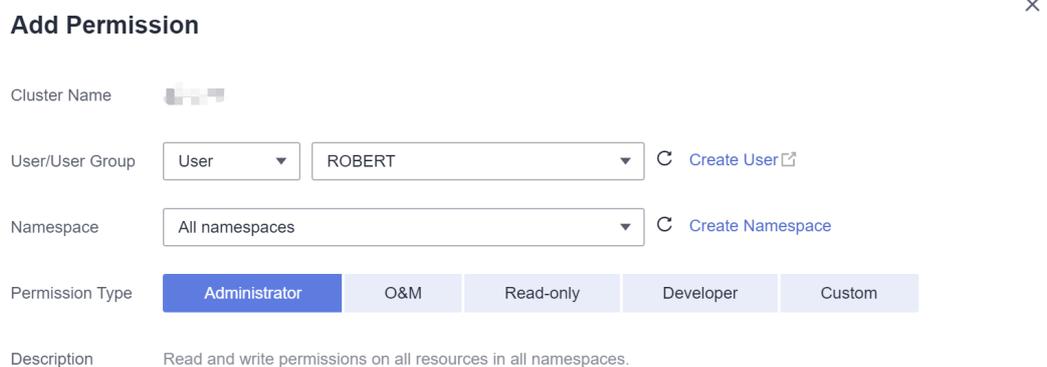
Una vez completada la configuración, James tiene los permisos de gestión de clústeres para todos los proyectos y los permisos de solo lectura en todos los espacios de nombres, y Robert, William, Linda y Peter tienen el permiso de solo lectura en todos los clústeres y espacios de nombres.

## Líder del equipo de desarrollo: Robert

En los pasos anteriores, a Robert se le ha asignado el permiso de solo lectura en todos los clústeres y espacios de nombres. Ahora, asigne los permisos de administrador en todos los espacios de nombres a Robert.

Por lo tanto, debe asignar los permisos de administrador en todos los espacios de nombres en todos los clústeres a Robert.

**Figura 16-15** Asignar permisos de administrador en espacios de nombres a Robert



**Add Permission** ×

Cluster Name [REDACTED]

User/User Group User ▼ ROBERT ▼ C Create User

Namespace All namespaces ▼ C Create Namespace

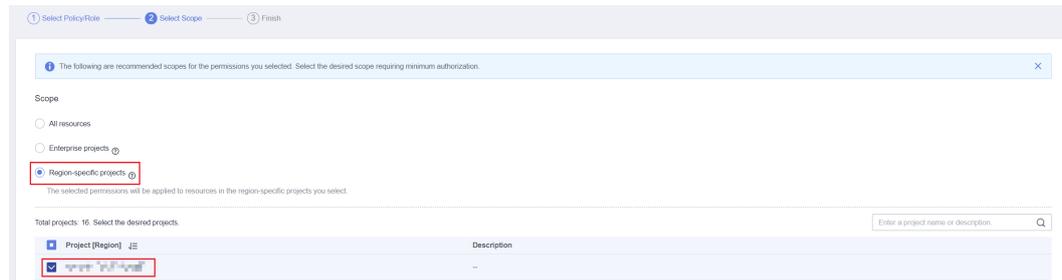
Permission Type Administrator O&M Read-only Developer Custom

Description Read and write permissions on all resources in all namespaces.

## Ingeniero de O&M: William

En los pasos anteriores, a William se le ha asignado el permiso de solo lectura en todos los clústeres y espacios de nombres. También requiere los permisos de gestión de clústeres en su región. Por lo tanto, puede iniciar sesión en la consola de IAM, crear un grupo de usuarios llamado **cce-sre-b4** y asignar CCE FullAccess a William para su región.

**Figura 16-16** Asignar los permisos de gestión de clústeres para la región Beijing4 al grupo de usuarios al que pertenece WILLIAM

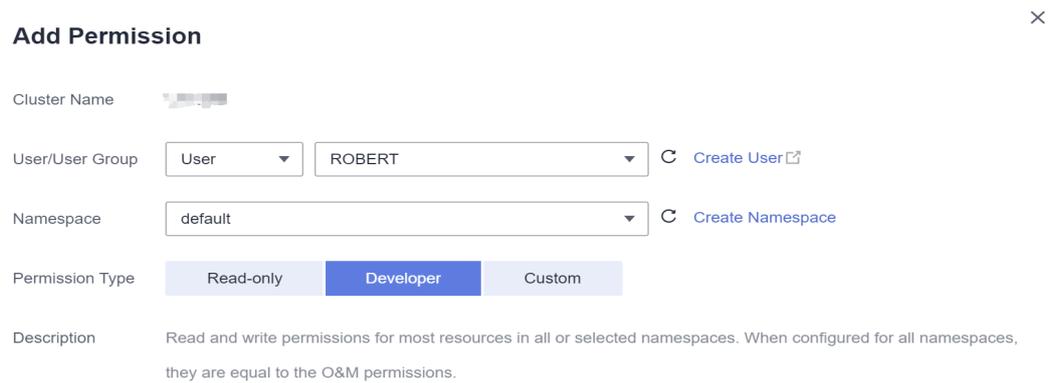


Ahora, William tiene los permisos de gestión de clúster para su región y el permiso de solo lectura en todos los espacios de nombres.

## Ingenieros de desarrollo: Linda y Peter

En los pasos anteriores, a Linda y Peter se les ha asignado el permiso de solo lectura en clústeres y espacios de nombres. Por lo tanto, solo necesita asignarles la política de edición.

**Figura 16-17** Asignación de la política de edición en espacios de nombres



Por ahora, todos los permisos requeridos se asignan a los miembros del departamento.

## 16.5 Dependencia de permisos de la consola de CCE

Algunas políticas de permisos de CCE dependen de las políticas de otros servicios en la nube. Para ver o usar otros recursos en la nube en la consola de CCE, debe habilitar la función de control de acceso de políticas del sistema de IAM y asignar políticas de dependencia para los otros servicios en la nube.

- Las políticas de dependencia se asignan según la política de FullAccess de CCE o de ReadOnlyAccess de CCE que configure. Para más detalles, consulte [Permisos de clúster \(basados en IAM\)](#).
- Solo los usuarios y grupos de usuarios con permisos de espacio de nombres pueden obtener el acceso de visualización a los recursos en clústeres de v1.11.7-r2 y posteriores.
  - Si se concede a un usuario el acceso de vista a todos los espacios de nombres de un clúster, el usuario puede ver todos los recursos de espacio de nombres (excepto los secretos) del clúster. Para ver secretos en el clúster, el usuario debe obtener el rol **admin** o **edit** en todos los espacios de nombres del clúster.

- Las políticas de CustomedHPA y de HPA solo tienen efecto después de que se hayan configurado los permisos de administrador del clúster para el espacio de nombres.
- El rol **view** dentro de un solo espacio de nombres permite a los usuarios ver recursos solo en el espacio de nombres especificado.

## Configuración de política de dependencia

Para conceder a un usuario de IAM los permisos para ver o usar recursos de otros servicios en la nube en la consola de CCE, primero debe conceder a la política de CCE Administrator, CCE FullAccess o CCE ReadOnlyAccess al grupo de usuarios al que pertenece el usuario y, a continuación, conceda al usuario las políticas de dependencia enumeradas en [Tabla 16-12](#). Estas políticas de dependencia permitirán al usuario de IAM acceder a los recursos de otros servicios en la nube.

### NOTA

**Enterprise projects** puede agrupar y gestionar recursos en diferentes proyectos de una empresa. Por lo tanto, los recursos quedan aislados. IAM le permite implementar la autorización de grano fino. Se recomienda encarecidamente que utilice IAM para la gestión de permisos.

Si utiliza un proyecto de empresa para establecer permisos para los usuarios de IAM, se aplicarán las siguientes restricciones:

- En la consola de CCE, los proyectos empresariales no pueden invocar a la API utilizada para obtener datos de supervisión de AOM para la supervisión de clústeres. Por lo tanto, los usuarios de IAM en estos proyectos de empresa no pueden consultar datos de supervisión.
- En la consola de CCE, los proyectos de empresa no pueden invocar a la API para consultar el par de claves creado durante la creación del nodo. Por lo tanto, los usuarios de IAM en estos proyectos de empresa no pueden usar el modo de inicio de sesión de par de claves. Solo se admite el modo de inicio de sesión con contraseña.
- En la consola de CCE, los proyectos de empresa no se admiten durante la creación de plantillas. Por lo tanto, los subusuarios de proyectos de empresa no pueden utilizar la gestión de plantillas.
- Después de asignar el permiso **CCE FullAccess** a un proyecto de empresa, debe configurar la acción **ecs:availabilityZones:list** en la consola de IAM para que los usuarios del proyecto de empresa puedan crear nodos. De lo contrario, el sistema le pedirá que no tenga el permiso.

CCE admite la configuración de permisos detallados, pero tiene las siguientes restricciones:

- AOM no admite la supervisión a nivel de recursos. Después de configurar los permisos de operación en recursos específicos mediante la función de gestión de recursos de clústeres de IAM, los usuarios de IAM pueden ver la información de supervisión de clústeres en la página **Panel** de la consola de CCE, pero no pueden ver los datos en métricas no de grano fino.

**Tabla 16-12** Políticas de dependencia

Función de consola	Servicios dependientes	Funciones o políticas requeridas
Panel	Application Operations Management (AOM)	<ul style="list-style-type: none"> <li>● Un usuario de IAM con el administrador de CCE asignado puede utilizar esta función solo después de que se asigne la política de FullAccess de AOM.</li> <li>● Los usuarios de IAM con IAM ReadOnlyAccess, CCE FullAccess o CCE ReadOnlyAccess asignados pueden utilizar directamente esta función.</li> </ul>
Gestión de la carga de trabajo	Elastic Load Balance (ELB) Application Performance Management (APM) Application Operations Management (AOM) NAT Gateway Object Storage Service (OBS) Scalable File Service (SFS)	Excepto en los siguientes casos, el usuario no requiere ningún rol adicional para crear cargas de trabajo. <ul style="list-style-type: none"> <li>● Para crear un Service mediante ELB, debe tener asignado el FullAccess ELB o el ELB Administrator más VPC Administrator.</li> <li>● Para utilizar un sondeo de Java, debe tener asignado el FullAccess AOM y el FullAccess APM.</li> <li>● Para crear un Service con NAT Gateway, debe tener asignado el NAT Gateway Administrator.</li> <li>● Para usar OBS, debe tener asignado el OBS Administrator globalmente.</li> </ul> <p><b>NOTA</b>                      Debido a la caché, se tarda unos 13 minutos para que la política de RBAC entre en vigor después de concederse a los usuarios, proyectos de empresa y grupos de usuarios. Después de que se concede una política de sistema relacionada con OBS, se tarda unos 5 minutos para que la política entre en vigor.</p> <ul style="list-style-type: none"> <li>● Para utilizar SFS, debe tener asignado el FullAccess de SFS.</li> </ul>
Gestión de clúster	Application Operations Management (AOM) Billing Center (BSS)	<ul style="list-style-type: none"> <li>● La expansión o la ampliación automáticas requiere la política de FullAccess de AOM.</li> <li>● Cambiar el modo de facturación a anual/mensual requiere el rol de BSS Administrator.</li> </ul>
Gestión de nodos	Elastic Cloud Server (ECS)	Si el permiso asignado a un usuario de IAM es CCE Administrator, la creación o eliminación de un nodo requiere la política FullAccess de ECS o ECS Administrator y la política de VPC Administrator.

Función de consola	Servicios dependientes	Funciones o políticas requeridas
Gestión de red	Elastic Load Balance (ELB) NAT Gateway	Excepto en los siguientes casos, el usuario no requiere ningún rol adicional para crear un Service. <ul style="list-style-type: none"> <li>● Para crear un Service mediante ELB, debe tener asignado el FullAccess ELB o el ELB Administrator más VPC Administrator.</li> <li>● Para crear un Service con NAT Gateway, debe tener asignado el NAT Administrator.</li> </ul>
Gestión de almacenamiento	Object Storage Service (OBS) Scalable File Service (SFS) SFS Turbo	<ul style="list-style-type: none"> <li>● Para usar OBS, debe tener asignado el OBS Administrator globalmente.</li> </ul> <p><b>NOTA</b>                      Debido a la caché, se tarda unos 13 minutos para que la política de RBAC entre en vigor después de concederse a los usuarios, proyectos de empresa y grupos de usuarios. Después de que se concede una política de sistema relacionada con OBS, se tarda unos 5 minutos para que la política entre en vigor.</p> <ul style="list-style-type: none"> <li>● Para utilizar SFS, debe tener asignado el FullAccess de SFS.</li> <li>● El uso de SFS Turbo requiere el rol de SFS Turbo Admin.</li> </ul> <p>El rol CCE Administrator es necesario para importar dispositivos de almacenamiento.</p>
Administración del espacio de nombres	/	/
Gestión de gráficos	/	Las cuentas en la nube y los usuarios de IAM con CCE Administrator asignado pueden usar esta función.
Gestión de complementos	/	Las cuentas en la nube y los usuarios de IAM con CCE Administrator, CCE FullAccess, or CCE ReadOnlyAccess asignados pueden usar esta función.

Función de consola	Servicios dependientes	Funciones o políticas requeridas
Gestión de permisos	/	<ul style="list-style-type: none"> <li>● Para las cuentas en la nube, no se requiere ninguna política o función adicional.</li> <li>● Los usuarios de IAM que tengan asignado el CCE Administrator o Security Administrator global pueden utilizar esta función.</li> <li>● Los usuarios de IAM con FullAccess CCE o ReadOnlyAccess CCE asignados pueden usar esta función.</li> </ul>
Centro de configuración	/	<ul style="list-style-type: none"> <li>● La creación de ConfigMaps no requiere ninguna política adicional.</li> <li>● La visualización de secretos requiere que el permiso de acluster-admin, admin o edit esté configurado para el espacio de nombres. Se debe asignar la política DEW KeypairFullAccess o DEW KeypairReadOnlyAccess a los servicios dependientes.</li> </ul>
Centro de ayuda	/	/
Cambio a otros servicios relacionados	Software Repository for Container (SWR) Application Operations Management (AOM) Multi-Cloud Container Platform (MCP)	La consola de CCE proporciona enlaces a otros servicios relacionados. Para ver o utilizar estos servicios, se deben asignar los permisos necesarios a un usuario de IAM para los servicios.

## 16.6 Seguridad del pod

### 16.6.1 Configuración de una política de seguridad de pod

Una política de seguridad de pod (PSP) es un recurso a nivel de clúster que controla los aspectos de seguridad confidenciales de la especificación de pod. El objeto de [PodSecurityPolicy](#) en Kubernetes define un grupo de condiciones que un pod debe cumplir para ser aceptado por el sistema, así como los valores predeterminados de los campos relacionados.

De forma predeterminada, el componente de control de acceso de PSP está habilitado para clústeres de v1.17.17 y se crea un PSP global predeterminado llamado **psp-global**. Puede modificar la política predeterminada (pero no eliminarla). También puede crear un PSP y vincularlo a la configuración de RBAC.

**NOTA**

- Además de la PSP predeterminada global, el sistema configura PSP independientes para los componentes del sistema en el espacio de nombres kube-system. La modificación de la configuración psp-global no afecta a la creación de pods en el espacio de nombres kube-system.
- En Kubernetes 1.25, PSP se ha eliminado y reemplazado por Admisión de seguridad para pods. Para obtener más información, véase [Configuración de admisión de seguridad de pods](#).

## Modificación de la PSP global predeterminada

Antes de modificar la PSP global predeterminada, asegúrese de que se ha creado y conectado un clúster de CCE mediante kubectl.

**Paso 1** Ejecute el siguiente comando:

```
kubectl edit psp psp-global
```

**Paso 2** Modifique los parámetros según sea necesario. Para obtener más información, consulte [PodSecurityPolicy](#).

----Fin

## Ejemplo de habilitación de Sysctls inseguros en la Política de Seguridad de Pods

Puede configurar los sistemas-unsafe-sysctls permitidos para un grupo de nodos. Para los clústeres de CCE de **v1.17.17** y versiones posteriores, agregue configuraciones en **allowedUnsafeSysctls** de la política de seguridad del pod para que la configuración surta efecto. Para obtener más información, consulte [PodSecurityPolicy](#).

Además de modificar la política de seguridad global de pods, puede agregar nuevas políticas de seguridad de pods. Por ejemplo, habilite sysctls inseguro **net.core.somaxconn**. A continuación se muestra un ejemplo de cómo agregar una política de seguridad de pod:

```
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  annotations:
    seccomp.security.alpha.kubernetes.io/allowedProfileNames: '*'
  name: sysctl-ppsp
spec:
  allowedUnsafeSysctls:
    - net.core.somaxconn
  allowPrivilegeEscalation: true
  allowedCapabilities:
    - '*'
  fsGroup:
    rule: RunAsAny
  hostIPC: true
  hostNetwork: true
  hostPID: true
  hostPorts:
    - max: 65535
      min: 0
  privileged: true
  runAsGroup:
    rule: RunAsAny
  runAsUser:
    rule: RunAsAny
  seLinux:
    rule: RunAsAny
  supplementalGroups:
    rule: RunAsAny
  volumes:
```

```

- '*'
---
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: sysctl-pp
rules:
  - apiGroups:
    - "*"
    resources:
    - podsecuritypolicies
    resourceName:
    - sysctl-pp
    verbs:
    - use
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: sysctl-pp
roleRef:
  kind: ClusterRole
  name: sysctl-pp
  apiGroup: rbac.authorization.k8s.io
subjects:
- kind: Group
  name: system:authenticated
  apiGroup: rbac.authorization.k8s.io
    
```

## Restauración de la PSP original

Si ha modificado la política de seguridad de pod predeterminada y desea restaurar la política de seguridad de pod original, realice las siguientes operaciones.

- Paso 1** Cree un archivo de descripción de política denominado **policy.yaml**. **policy.yaml** es un nombre de archivo de ejemplo. Puede cambiar el nombre según sea necesario.

### vi policy.yaml

El contenido del archivo de descripción es el siguiente:

```

apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: psp-global
  annotations:
    seccomp.security.alpha.kubernetes.io/allowedProfileNames: '*'
spec:
  privileged: true
  allowPrivilegeEscalation: true
  allowedCapabilities:
  - '*'
  volumes:
  - '*'
  hostNetwork: true
  hostPorts:
  - min: 0
    max: 65535
  hostIPC: true
  hostPID: true
  runAsUser:
    rule: 'RunAsAny'
  seLinux:
    rule: 'RunAsAny'
  supplementalGroups:
    rule: 'RunAsAny'
  fsGroup:
    
```

```

    rule: 'RunAsAny'
---
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: psp-global
rules:
  - apiGroups:
    - "*"
    resources:
    - podsecuritypolicies
    resourceName:
    - psp-global
    verbs:
    - use
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: psp-global
roleRef:
  kind: ClusterRole
  name: psp-global
  apiGroup: rbac.authorization.k8s.io
subjects:
- kind: Group
  name: system:authenticated
  apiGroup: rbac.authorization.k8s.io
    
```

**Paso 2** Ejecute el siguiente comando:

**kubectl apply -f policy.yaml**

**----Fin**

## 16.6.2 Configuración de admisión de seguridad de pods

Antes de usar la [admisión de seguridad de pods](#), debe comprender los [Estándares de seguridad de Pod](#) de Kubernetes. Estos estándares definen diferentes niveles de aislamiento para los pods. Te permiten definir cómo quieres restringir el comportamiento de los pods de una manera clara y consistente. Kubernetes ofrece un controlador de admisión de seguridad de pod incorporado para hacer cumplir los estándares de seguridad de pod. Las restricciones de seguridad de los pods se aplican a nivel de espacio de nombres cuando se crean los pods.

El estándar de seguridad de pod define tres niveles de política de seguridad:

**Tabla 16-13** Niveles de política de seguridad de pod

Nivel	Descripción
privileged	Política sin restricciones, que proporciona el nivel más amplio posible de permisos, generalmente dirigidos a cargas de trabajo a nivel de sistema e infraestructura administradas por usuarios privilegiados y de confianza, como CNI y controladores de almacenamiento.
baseline	Política mínimamente restrictiva que evita las escalaciones de privilegios conocidas, normalmente dirigidas a cargas de trabajo no críticas. Esta política deshabilita capacidades como hostNetwork y hostPID.
restricted	Política muy restringida, siguiendo las mejores prácticas actuales de endurecimiento de Pod.

La **admisión de seguridad de pod** se aplica a nivel de espacio de nombres. El controlador restringe el contexto de seguridad y otros parámetros en el pod o contenedor en el espacio de nombres. La política de privilegios no verifica el campo **securityContext** del pod y el contenedor. La línea de base y las políticas restringidas tienen requisitos diferentes sobre **securityContext**. Para obtener más información, consulte **Normas de seguridad de Pod**.

Configuración del contexto de seguridad: **Configurar un contexto de seguridad para un pod o contenedor**

## Etiquetas de admisión de seguridad de pod

Kubernetes define tres tipos de etiquetas para la admisión de seguridad de pods (consulte **Tabla 16-14**). Puede establecer estas etiquetas en un espacio de nombres para definir el nivel estándar de seguridad del pod que se utilizará. Sin embargo, no cambie el nivel estándar de seguridad pod en espacios de nombres del sistema como kube-system. De lo contrario, los pods en el espacio de nombres del sistema pueden estar defectuosos.

**Tabla 16-14** Etiquetas de admisión de seguridad de pod

Modo	Objeto de destino	Descripción
enforce	Pods	Las violaciones de la política provocarán que el pod sea rechazado.
audit	Cargas de trabajo (como la Deployment y el trabajo)	Las violaciones de políticas activarán la adición de una anotación de auditoría al evento registrado en el log de auditoría, pero se permiten de otro modo.
warn	Cargas de trabajo (como la Deployment y el trabajo)	Las violaciones de políticas activarán una advertencia de cara al usuario, pero están permitidas de otro modo.

### NOTA

Los pods a menudo se crean indirectamente, mediante la creación de un objeto de carga de trabajo, como una Deployment o un trabajo. Para ayudar a detectar las infracciones con anticipación, tanto los modos de auditoría como de advertencia se aplican a los recursos de la carga de trabajo. Sin embargo, el modo de aplicación se aplica solo a los objetos pod resultantes.

## Hacer cumplir la admisión de seguridad de pod con etiquetas de espacio de nombres

Puede etiquetar espacios de nombres para aplicar estándares de seguridad de pod. Suponga que un espacio de nombres está configurado de la siguiente manera:

```
apiVersion: v1
kind: Namespace
metadata:
  name: my-baseline-namespace
  labels:
```

```
pod-security.kubernetes.io/enforce: privileged
pod-security.kubernetes.io/enforce-version: v1.25
pod-security.kubernetes.io/audit: baseline
pod-security.kubernetes.io/audit-version: v1.25
pod-security.kubernetes.io/warn: restricted
pod-security.kubernetes.io/warn-version: v1.25

# The label can be in either of the following formats:
# pod-security.kubernetes.io/<MODE>: <LEVEL>
# pod-security.kubernetes.io/<MODE>-version: <VERSION>
# The audit and warn modes inform you of which security behaviors are
violated by the load.
```

Las etiquetas del espacio de nombres indican el nivel de política que se aplicará al modo. Para cada modo, hay dos etiquetas que determinan la política utilizada:

- `pod-security.kubernetes.io/<MODE>: <LEVEL>`
  - `<MODE>`: debe ser **enforce**, **audit** o **warn**. Para obtener más información sobre los modos, consulte [Tabla 16-14](#).
  - `<LEVEL>`: debe ser **privileged**, **baseline** o **restricted**. Para obtener más información sobre los niveles, consulte [Tabla 16-13](#).
- `pod-security.kubernetes.io/<MODE>-version: <VERSION>`

Opcional, que ancla la política a una versión determinada de Kubernetes.

  - `<MODE>`: debe ser **enforce**, **audit** o **warn**. Para obtener más información sobre los modos, consulte [Tabla 16-14](#).
  - `<VERSION>`: número de versión de Kubernetes. Por ejemplo, `v1.25`. También puede utilizar **latest**.

Si los pods se despliegan en el espacio de nombres anterior, se aplican las siguientes restricciones de seguridad:

1. La verificación en el modo de aplicación se omite (modo de enforce + nivel de privileged).
2. Se verifican las restricciones relacionadas con la política de línea de base (modo de audit + nivel de baseline). Es decir, si el pod o contenedor infringen la política, el evento correspondiente se registra en el log de auditoría.
3. Se verifican las restricciones relacionadas con la política restringida (modo de warn + nivel restricted). Es decir, si el pod o contenedor infringen la política, el usuario recibirá una alarma al crear el pod.

## Migración de la política de seguridad de pods a admisión de seguridad de pods

Si utiliza políticas de seguridad de pods en un clúster anterior a `v1.25` y necesita reemplazarlas con admisión de seguridad de pods en un clúster de `v1.25` o posterior, siga la guía de [Migración de PodSecurityPolicy a la controladora de admisión integrada de PodSecurity](#).

## AVISO

1. La admisión de seguridad de pod admite solo tres modos de aislamiento, menos flexible que las políticas de seguridad de pod. Si necesita más control sobre restricciones específicas, necesitará usar un Validating Admission Webhook para aplicar esas políticas.
2. La admisión de seguridad de pod es un controlador de admisión que no cambia, lo que significa que no modificará los pods antes de validarlos. Si confiaba en este aspecto de PSP, necesitará modificar el contexto de seguridad en sus cargas de trabajo, o usar un Mutating Admission Webhook para realizar esos cambios.
3. PSP le permite vincular diferentes políticas a diferentes cuentas de servicio. Este enfoque tiene muchas trampas y no se recomienda, pero si requiere esta función de todos modos, tendrá que usar un webhook de terceros en su lugar.
4. No aplique la admisión de seguridad de pod a los espacios de nombres donde se despliegan los componentes de CCE, como kube-system, kube-public y kube-node-lease. De lo contrario, los componentes de CCE y las funciones adicionales serán anormales.

## Documentación

- [Admisión de seguridad de pod](#)
- [Asignación de PodSecurityPolicies a estándares de seguridad de pod](#)
- [Aplique estándares de seguridad de pods con etiquetas de espacio de nombres](#)
- [Aplique los estándares de seguridad de pod mediante la configuración del controlador de admisión integrado](#)

## 16.7 Mejora de la seguridad del token de la cuenta de Service

En clústeres anteriores a v1.21, se obtiene un token montando el secreto de la cuenta de servicio en un pod. Los tokens obtenidos de esta manera son permanentes. Este enfoque ya no se recomienda a partir de la versión 1.21. Las cuentas de Service detendrán la creación automática de secretos en clústeres a partir de la versión 1.25.

En clústeres de la versión 1.21 o posterior, puede usar la API [TokenRequest](#) para obtener el token y usar el volumen proyectado para montar el token en el pod. Dichos tokens son válidos por un período fijo (una hora por defecto). Antes de la expiración, Kubelet actualiza el token para asegurarse de que el pod siempre utiliza un token válido. Cuando se elimina la cápsula de montaje, el token se invalida automáticamente. Este enfoque se implementa mediante la función [BoundServiceAccountTokenVolume](#) para mejorar la seguridad del token de la cuenta de servicio. Los clústeres de v1.21 o posterior habilitan este enfoque de forma predeterminada.

Para una transición sin problemas, la comunidad extiende el período de validez del token a un año por defecto. Después de un año, el token no es válido y los clientes que no admiten la recarga de certificados no pueden acceder al servidor API. Se recomienda que los clientes de versiones anteriores se actualicen lo antes posible. De lo contrario, pueden producirse fallas en el servicio.

Si utiliza un cliente de Kubernetes de una versión pendiente de actualización, la recarga del certificado puede fallar. Las versiones de las bibliotecas cliente de Kubernetes oficialmente admitidas que pueden recargar tokens son las siguientes:

- Go:  $\geq$  v0.15.7
- Python:  $\geq$  v12.0.0
- Java:  $\geq$  v9.0.0
- Javascript:  $\geq$  v0.10.3
- Ruby: master branch
- Haskell: v0.3.0.0
- C#:  $\geq$  7.0.5

Para obtener más información, visite <https://github.com/kubernetes/enhancements/tree/master/keps/sig-auth/1205-bound-service-account-tokens>.

#### NOTA

Si necesita un token que nunca caduca, también puede [gestionar manualmente los secretos de las cuentas de servicio](#). Aunque se puede crear manualmente un token de cuenta de servicio permanente, se recomienda utilizar un token de corta duración llamando a la API `TokenRequest` para mayor seguridad.

## Diagnóstico

Realice los siguientes pasos para comprobar los clústeres de CCE de v1.21 o posterior:

1. Compruebe las versiones adicionales.
  - Si está utilizando el complemento prometheus v2.23.34 o anterior, actualícelo a v2.23.34 o posterior.
  - Si está utilizando el complemento npd v1.15.0 o anterior, actualícelo a la versión más reciente.
2. Utilice `kubectl` para conectarse al clúster y ejecute el comando `kubectl get --raw "/metrics" | grep stale obsoleto` para consultar las métricas. Compruebe la métrica denominada `serviceaccount_stale_tokens_total`.

Si el valor es mayor que 0, algunas cargas de trabajo del clúster pueden estar utilizando una versión anterior de `client-go`. En este caso, compruebe si este problema se produce en las aplicaciones desplegadas. En caso afirmativo, actualice `client-go` a la versión especificada por la comunidad lo antes posible. La versión debe ser al menos dos versiones principales del clúster de CCE. Por ejemplo, si la versión de clúster es 1.23, la versión de la biblioteca de dependencias de Kubernetes debe ser al menos 1.19.

```
[root@ ~]# kubectl get --raw "/metrics" | grep stale
# HELP serviceaccount_stale_tokens_total [ALPHA] Cumulative stale projected service account tokens used
# TYPE serviceaccount_stale_tokens_total counter
serviceaccount_stale_tokens_total 52
```

## 16.8 Descripción de la confiabilidad del sistema

CCE trabaja estrechamente con múltiples servicios en la nube para admitir las funciones informáticas, de almacenamiento, de redes y de supervisión. Cuando inicia sesión en la consola de CCE por primera vez, CCE solicita automáticamente permisos para acceder a los servicios en la nube en la región donde ejecuta sus aplicaciones. En especial:

- Servicios de cómputo

Cuando se crea un nodo en un clúster, se crea un servidor en la nube en consecuencia. El requisito previo es que CCE haya obtenido los permisos para acceder a Elastic Cloud Service (ECS) y Bare Metal Server (BMS).

- Servicios de almacenamiento  
CCE permite montar el almacenamiento en nodos y contenedores de un clúster. El requisito previo es que CCE haya obtenido los permisos para acceder a servicios como Elastic Volume Service (EVS), Scalable File Service (SFS) y Object Storage Service (OBS).
- Servicios de redes  
CCE permite que los contenedores de un clúster se publiquen como servicios a los que pueden acceder los sistemas externos. El requisito previo es que CCE haya obtenido los permisos para acceder a servicios como Virtual Private Cloud (VPC) y Elastic Load Balance (ELB).
- Servicios de contenedores y monitorización  
CCE admite funciones como la extracción de imágenes de contenedores, la supervisión y el registro. El requisito previo es que CCE haya obtenido los permisos para acceder a servicios como SoftWare Repository for Container (SWR) y Application Operations Management (AOM).

Después de aceptar la atribución, CCE crea automáticamente una delegación en IAM para delegar otros permisos de operación de recursos en su cuenta a Huawei Cloud CCE. Para obtener más información, consulte [Delegación de la cuenta](#).

Las agencias creadas automáticamente por CCE son las siguientes:

- [cce\\_admin\\_trust](#)
- [cce\\_cluster\\_agency](#)

## cce\_admin\_trust

La delegación `cce_admin_trust` tiene los permisos de Tenant Administrator. Tenant Administrator tiene los permisos en todos los servicios en la nube excepto IAM, que se utilizan para invocar a los servicios en la nube de los que depende CCE. La delegación solo tiene efecto en la región actual.

Para usar CCE en varias regiones, debe solicitar permisos de recursos en la nube en cada región. Puede ir a la consola de IAM, elegir **Agencies** y hacer clic en `cce_admin_trust` para ver los registros de delegación de cada región.

### NOTA

CCE puede no ejecutarse como se esperaba si el rol Tenant Administrator no está asignado. Por lo tanto, no elimine ni modifique la delegación `cce_admin_trust` cuando utilice CCE.

## cce\_cluster\_agency

La delegación `cce_cluster_agency` contiene solo los permisos de operación de recursos de servicio en la nube requeridos por los componentes de CCE. Genera credenciales de acceso temporales utilizadas por los componentes en clústeres de CCE.

### NOTA

- La delegación `cce_cluster_agency` admite clústeres de v1.21 o posterior.
- Cuando se crea la delegación `cce_cluster_agency`, se crea automáticamente una política personalizada llamada **CCE cluster policies**. No elimine esta política.

Si los permisos de la delegación `cce_cluster_agency` son diferentes de los esperados por CCE, la consola muestra un mensaje indicando que los permisos han cambiado y que necesita volver a autorizar la delegación.



La delegación `cce_cluster_agency` puede ser reautorizada en los siguientes escenarios:

- Los permisos de los que dependen los componentes de CCE pueden cambiar con las versiones. Por ejemplo, si un nuevo componente depende de nuevos permisos, CCE actualizará la lista de permisos esperados y deberá conceder permisos a `cce_cluster_agency` de nuevo.
- Cuando modifica manualmente los permisos de la delegación `cce_cluster_agency`, los permisos de la delegación son diferentes de los esperados por CCE. En este caso, se muestra un mensaje, pidiéndole que vuelva a autorizar a la delegación. Si vuelve a autorizar la delegación, los permisos modificados manualmente pueden no ser válidos.

# 17 Cloud Trace Service (CTS)

## 17.1 Operaciones de CCE con el apoyo de CTS

Cloud Trace Service (CTS) registra las operaciones en recursos de servicios en la nube, lo que permite a los usuarios consultar, auditar y realizar un seguimiento de las solicitudes de operación de recursos iniciadas desde la consola de gestión o las API abiertas, así como las respuestas a las solicitudes.

**Tabla 17-1** Operaciones de CCE con el apoyo de CTS

Operación	Tipo de recurso	Nombre del evento
Creación de una delegación	Clúster	createUserAgencies
Creación de un clúster	Clúster	createCluster
Creación de un clúster de facturación anual/mensual	Clúster	createCluster/createPeriodicCluster
Actualización de la descripción de un clúster	Clúster	updateCluster
Actualización de un clúster	Clúster	clusterUpgrade
Eliminación de un clúster	Clúster	claimCluster/deleteCluster
Descarga de un certificado de clúster	Clúster	getClusterCertByUID
Vinculación y desvinculación de una EIP	Clúster	operateMasterEIP

Operación	Tipo de recurso	Nombre del evento
Despertar un clúster y restablecer la gestión de nodos (V2)	Clúster	operateCluster
Hibernación de un clúster (V3)	Clúster	hibernateCluster
Despertar un clúster (V3)	Clúster	awakeCluster
Cambio de las especificaciones de un clúster de pago por uso	Clúster	resizeCluster
Cambio de las especificaciones de un clúster de facturación anual/mensual	Clúster	resizePeriodCluster
Modificación de las configuraciones de un clúster	Clúster	updateConfiguration
Creación de un grupo de nodos	Grupo de nodos	createNodePool
Actualización de un grupo de nodos	Grupo de nodos	updateNodePool
Eliminación de un grupo de nodos	Grupo de nodos	claimNodePool
Migración de un grupo de nodos	Grupo de nodos	migrateNodepool
Modificación de configuraciones de grupo de nodos	Grupo de nodos	updateConfiguration
Creación de un nodo	Nodo	createNode
Creación de un nodo de facturación anual/mensual	Nodo	createPeriodNode
Eliminación de todos los nodos de un clúster especificado	Nodo	deleteAllHosts
Eliminación de un solo nodo	Nodo	deleteOneHost/claimOneHost
Actualización de la descripción de un nodo	Nodo	updateNode

Operación	Tipo de recurso	Nombre del evento
Creación de una instancia de complemento	Instancia de complementos	createAddonInstance
Eliminación de una instancia de complemento	Instancia de complementos	deleteAddonInstance
Carga de un gráfico	Gráfico	uploadChart
Actualización de un gráfico	Gráfico	updateChart
Eliminación de un gráfico	Gráfico	deleteChart
Creación de una versión	Lanzamiento	createRelease
Actualización de una versión	Lanzamiento	updateRelease
Eliminación de una versión	Lanzamiento	deleteRelease
Creación de un ConfigMap	Recursos de Kubernetes	createConfigmaps
Creación de un DaemonSet	Recursos de Kubernetes	createDaemonsets
Creación de un Deployment	Recursos de Kubernetes	createDeployments
Creación de un evento	Recursos de Kubernetes	createEvents
Creación de un ingreso	Recursos de Kubernetes	createIngresses
Creación de un trabajo	Recursos de Kubernetes	createJobs
Creación de un espacio de nombres	Recursos de Kubernetes	createNamespaces
Creación de un nodo	Recursos de Kubernetes	createNodes
Creación de un PersistentVolumeClaim	Recursos de Kubernetes	createPersistentvolumeclaims
Creación de un pod	Recursos de Kubernetes	createPods
Creación de un conjunto de réplicas	Recursos de Kubernetes	createReplicasets

Operación	Tipo de recurso	Nombre del evento
Creación de una cuota de recursos	Recursos de Kubernetes	createResourcequotas
Creación de un secreto	Recursos de Kubernetes	createSecrets
Creación de un servicio	Recursos de Kubernetes	createServices
Creación de un StatefulSet	Recursos de Kubernetes	createStatefulsets
Creación de un volumen	Recursos de Kubernetes	createVolumes
Eliminación de un ConfigMap	Recursos de Kubernetes	deleteConfigmaps
Eliminación de un DaemonSet	Recursos de Kubernetes	deleteDaemonsets
Eliminación de un Deployment	Recursos de Kubernetes	deleteDeployments
Eliminación de un evento	Recursos de Kubernetes	deleteEvents
Eliminación de un ingreso	Recursos de Kubernetes	deleteIngresses
Eliminación de un trabajo	Recursos de Kubernetes	deleteJobs
Eliminación de un espacio de nombres	Recursos de Kubernetes	deleteNamespaces
Eliminación de un nodo	Recursos de Kubernetes	deleteNodes
Eliminación de un pod	Recursos de Kubernetes	deletePods
Eliminación de un conjunto de réplicas	Recursos de Kubernetes	deleteReplicasets
Supresión de una cuota de recursos	Recursos de Kubernetes	deleteResourcequotas
Eliminación de un secreto	Recursos de Kubernetes	deleteSecrets
Eliminación de un servicio	Recursos de Kubernetes	deleteServices
Eliminación de un StatefulSet	Recursos de Kubernetes	deleteStatefulsets

Operación	Tipo de recurso	Nombre del evento
Supresión de volúmenes	Recursos de Kubernetes	deleteVolumes
Reemplazar un ConfigMap especificado	Recursos de Kubernetes	updateConfigmaps
Reemplazar un DaemonSet especificado	Recursos de Kubernetes	updateDaemonsets
Reemplazar una Deployment especificada	Recursos de Kubernetes	updateDeployments
Reemplazar un evento especificado	Recursos de Kubernetes	updateEvents
Reemplazar un ingreso especificado	Recursos de Kubernetes	updateIngresses
Reemplazar un trabajo especificado	Recursos de Kubernetes	updateJobs
Reemplazar un espacio de nombres especificado	Recursos de Kubernetes	updateNamespaces
Reemplazar un nodo especificado	Recursos de Kubernetes	updateNodes
Reemplazar un PersistentVolumeClaim especificado	Recursos de Kubernetes	updatePersistentvolumeclaims
Reemplazar un pod especificado	Recursos de Kubernetes	updatePods
Reemplazar un conjunto de réplicas especificado	Recursos de Kubernetes	updateReplicasets
Reemplazar una cuota de recursos especificada	Recursos de Kubernetes	updateResourcequotas
Reemplazar un secreto especificado	Recursos de Kubernetes	updateSecrets
Reemplazar un servicio especificado	Recursos de Kubernetes	updateServices
Reemplazar un StatefulSet especificado	Recursos de Kubernetes	updateStatefulsets

Operación	Tipo de recurso	Nombre del evento
Reemplazar el estado especificado	Recursos de Kubernetes	updateStatus
Carga de un gráfico	Recursos de Kubernetes	uploadChart
Actualización de una plantilla de componente	Recursos de Kubernetes	updateChart
Eliminación de un gráfico	Recursos de Kubernetes	deleteChart
Creación de una aplicación de plantilla	Recursos de Kubernetes	createRelease
Actualización de una aplicación de plantilla	Recursos de Kubernetes	updateRelease
Eliminación de una aplicación de plantilla	Recursos de Kubernetes	deleteRelease

## 17.2 Consulta de logs de CTS

### Escenario

Después de habilitar CTS, el sistema inicia las operaciones de grabación en los recursos de CCE. Los registros de operación de los últimos 7 días se pueden ver en la consola de gestión de CTS.

### Procedimiento

- Paso 1** Inicie sesión en la consola de gestión.
- Paso 2** Haga clic en  en la esquina superior izquierda y seleccione una región.
- Paso 3** Elija **Service List** en el menú principal. Elija **Management & Deployment > Cloud Trace Service**.
- Paso 4** En el panel de navegación de la consola de CTS, elija **Cloud Trace Service > Trace List**.
- Paso 5** En la página **Trace List**, consulte los registros de operaciones según los criterios de búsqueda. Actualmente, la lista de seguimiento admite consultas de seguimiento basadas en la combinación de los siguientes criterios de búsqueda:
  - **Trace Source, Resource Type, y Search By**  
 Seleccione los criterios de búsqueda de las listas desplegables. Seleccione **CCE** en la lista desplegable **Trace Source**.  
 Si selecciona **Trace name** en la lista desplegable **Search By**, especifique el nombre de seguimiento.  
 Si selecciona **Resource ID** en la lista desplegable **Search By**, seleccione o introduzca un ID de recurso específico.

Si selecciona **Resource name** en la lista desplegable **Search By**, seleccione o introduzca un nombre de recurso específico.

- **Operator:** Seleccione un operador específico (a nivel de usuario en lugar de a nivel de cuenta).
- **Trace Status:** Establezca este parámetro en cualquiera de los siguientes valores: **All trace statuses**, **normal**, **warning** y **incident**.
- **Intervalo de tiempo:** Puede consultar las trazas generadas durante cualquier intervalo de tiempo en los últimos siete días.

**Paso 6** Haga clic en  en la izquierda de un trazo para ampliar sus detalles, como se muestra a continuación.

**Figura 17-1** Ampliación de los detalles de seguimiento

Trace Name	Resource Ty...	Trace So...	Resource ID ⓘ	Resource Na...	Trace Stat...	Operator ⓘ	Operation Time	Operation
operateClus...	clusters-acti...	CCE			<span style="color: green;">✔</span> normal	xuchun...	Oct 11, 2018 09:24:56 GMT...	<a href="#">View Trace</a>
Trace ID	7660c4e0-ccf4-11e8-9fe5-286ed488cbe3			Source IP Address	58.213.108.72			
Trace Type	ConsoleAction			Generated	Oct 11, 2018 09:24:56 GMT+08:00			

**Paso 7** Haga clic en **View Trace** en la columna **Operation**. Se muestran los detalles de seguimiento.

**Figura 17-2** Consulta de los detalles del evento

```
{
  "service_type": "CCE",
  "user": {
    "domain": {
      "name": "xuchun@huawei.com",
      "id": "cc9c0076188843ea938743dd166c7fef"
    },
    "id": "69d8a0dc65e2436eb973093b49bb5ecb",
    "name": "xuchun@huawei.com"
  },
  "time": "2018/10/11 09:24:56 GMT+08:00",
  "code": 200,
  "resource_type": "clusters-action",
  "source_ip": "58.213.108.72",
  "trace_name": "operateCluster",
  "trace_type": "ConsoleAction",
  "message": "UserName: xuchun@huawei.com; Operation : POST /api/v2/projects/6244f46518ab48d4ba84b3bcb93aba67/clusters/3cc...",
  "record_time": "2018/10/11 09:24:56 GMT+08:00",
  "trace_id": "7660c4e0-ccf4-11e8-9fe5-286ed488cbe3",
  "trace_status": "normal"
}
```

----Fin

# 18 Gestión del almacenamiento: FlexVolume (desusado)

## 18.1 Descripción de FlexVolume

En el almacenamiento de contenedor, puede utilizar diferentes tipos de volúmenes y montarlos en contenedores en pods tantos como desee.

En CCE, el almacenamiento de contenedor está respaldado tanto por objetos nativos de Kubernetes, como `emptyDir`, `hostPath`, `secret` y `ConfigMap` como por servicios de almacenamiento en la nube.

Los clústeres CCE de **1.13 y versiones anteriores** utilizan el complemento **storage-driver** para conectarse a los servicios de almacenamiento a admitir el controlador de FlexVolume de Kubernetes para el almacenamiento de contenedor. El controlador de FlexVolume ha sido obsoleto a favor de la Container Storage Interface (CSI). **El complemento de everest para CSI se instala en los clústeres de CCE de las versiones 1.15 y posteriores de forma predeterminada.** Para obtener más información, véase [Descripción del almacenamiento de contenedores](#).

### NOTA

- En los clústeres de CCE anteriores a Kubernetes 1.13, no se admite la expansión de la capacidad extremo a extremo del almacenamiento de contenedor y la capacidad de PVC es incompatible con la capacidad de almacenamiento.
- **En un clúster de v1.13 o anteriores**, cuando una actualización o corrección de errores está disponible para las funcionalidades de almacenamiento, solo necesita instalar o actualizar el complemento del storage-driver. No es necesario actualizar el clúster ni crear un clúster.

## Restricciones

- Para clústeres creados en CCE, Kubernetes v1.15.11 es una versión transitoria en la que el complemento FlexVolume (**storage-driver**) es compatible con el complemento CSI (**everest**). Los clústeres de v1.17 y versiones posteriores ya no son compatibles con FlexVolume. Necesita usar el complemento más antiguo.
- El complemento FlexVolume será mantenido por los desarrolladores de Kubernetes, pero las nuevas funcionalidades solo se agregarán a CSI. Se aconseja no crear más almacenamiento que se conecte al complemento FlexVolume (storage-driver) en CCE. De lo contrario, los recursos de almacenamiento pueden no funcionar normalmente.

## Comprobación de complementos de almacenamiento

- Paso 1** Inicie sesión en la consola de CCE.
- Paso 2** En el árbol de navegación de la izquierda, haga clic en **Add-ons**.
- Paso 3** Haga clic en la ficha **Add-on Instance**.
- Paso 4** Seleccione un clúster en la esquina superior derecha. Se muestra el complemento de almacenamiento predeterminado instalado durante la creación del clúster.

----Fin

## Differences Between CSI and FlexVolume Plug-ins

**Tabla 18-1** CSI and FlexVolume

Kubernete s Solution	CCE Add-on	Feature	Recommendation
CSI	Everest	<p>CSI was developed as a standard for exposing arbitrary block and file storage storage systems to containerized workloads. Using CSI, third-party storage providers can deploy plugins exposing new storage systems in Kubernetes without having to touch the core Kubernetes code. In CCE, the everest add-on is installed by default in clusters of Kubernetes v1.15 and later to connect to storage services (EVS, OBS, SFS, and SFS Turbo).</p> <p>The everest add-on consists of two parts:</p> <ul style="list-style-type: none"> <li>● <b>everest-csi-controller</b> for storage volume creation, deletion, capacity expansion, and cloud disk snapshots</li> <li>● <b>everest-csi-driver</b> for mounting, unmounting, and formatting storage volumes on nodes</li> </ul> <p>For details, see <a href="#">everest</a>.</p>	<p>The <a href="#">everest</a> add-on is installed by default in clusters of <b>v1.15 and later</b>. CCE will mirror the Kubernetes community by providing continuous support for updated CSI capabilities.</p>

Kubernete s Solution	CCE Add-on	Feature	Recommendation
Flexvolume	storage- driver	FlexVolume is an out-of-tree plugin interface that has existed in Kubernetes since version 1.2 (before CSI). CCE provided FlexVolume volumes through the storage-driver add-on installed in clusters of Kubernetes v1.13 and earlier versions. This add-on connects clusters to storage services (EVS, OBS, SFS, and SFS Turbo).  For details, see <a href="#">storage-driver</a> .	For the created clusters of <b>v1.13 or earlier</b> , the installed FlexVolume plug-in (CCE add-on <b>storage-driver</b> ) can still be used. CCE stops providing update support for this add-on, and you are advised to <b>upgrade these clusters</b> .

**NOTA**

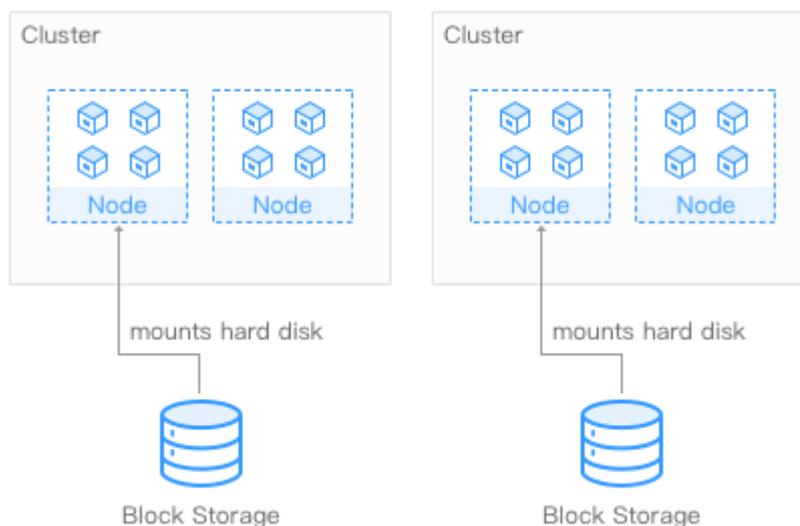
- A cluster can use only one type of storage plug-ins.
- The FlexVolume plug-in cannot be replaced by the CSI plug-in in clusters of v1.13 or earlier. You can only upgrade these clusters. For details, see [Actualización de clúster](#).

## 18.2 Using EVS Disks as Storage Volumes

### 18.2.1 Overview

To achieve persistent storage, CCE allows you to mount the storage volumes created from Elastic Volume Service (EVS) disks to a path of a container. When the container is migrated, the mounted EVS volumes are also migrated. By using EVS volumes, you can mount the remote file directory of storage system into a container so that data in the data volume is permanently preserved even when the container is deleted.

**Figura 18-1** Mounting EVS volumes to CCE



## Description

- **User-friendly:** Similar to formatting disks for on-site servers in traditional layouts, you can format block storage (disks) mounted to cloud servers, and create file systems on them.
- **Data isolation:** Each server uses an independent block storage device (disk).
- **Private network:** User can access data only in private networks of data centers.
- **Capacity and performance:** The capacity of a single volume is limited (TB-level), but the performance is excellent (ms-level read/write I/O latency).
- **Restriction:** EVS disks that have partitions or have non-ext4 file systems cannot be imported.
- **Applications:** HPC, enterprise core applications running in clusters, enterprise application systems, and development and testing. These volumes are often used by single-pod Deployments and jobs, or exclusively by each pod in a StatefulSet. EVS disks are non-shared storage and cannot be attached to multiple nodes at the same time. If two pods are configured to use the same EVS disk and the two pods are scheduled to different nodes, one pod cannot be started because the EVS disk cannot be attached to it.

## 18.2.2 (kubectl) Automatically Creating an EVS Disk

### Notes and Constraints

The following configuration example applies to clusters of Kubernetes 1.13 or earlier.

### Procedure

**Paso 1** Use kubectl to connect to the cluster. For details, see [Conexión a un clúster con kubectl](#).

**Paso 2** Run the following commands to configure the `pvc-evs-auto-example.yaml` file, which is used to create a PVC.

```
touch pvc-evs-auto-example.yaml
```

```
vi pvc-evs-auto-example.yaml
```

**Example YAML file for clusters of v1.9, v1.11, and v1.13:**

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-evs-auto-example
  namespace: default
  annotations:
    volume.beta.kubernetes.io/storage-class: sas
  labels:
    failure-domain.beta.kubernetes.io/region: ap-southeast-1
    failure-domain.beta.kubernetes.io/zone: ap-southeast-1a
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
```

**Tabla 18-2** Key parameters

Parameter	Description
volume.beta.kubernetes.io/storage-class	EVS disk type. The value is in lowercase.
failure-domain.beta.kubernetes.io/region	Region where the cluster is located.
failure-domain.beta.kubernetes.io/zone	AZ where the EVS volume is created. It must be the same as the AZ planned for the workload.
storage	Storage capacity in the unit of Gi.
accessModes	Read/write mode of the volume. You can set this parameter to <b>ReadWriteMany</b> (shared volume) and <b>ReadWriteOnce</b> (non-shared volume).

**Paso 3** Run the following command to create a PVC.

```
kubectl create -f pvc-evs-auto-example.yaml
```

After the command is executed, an EVS disk is created in the partition where the cluster is located. Choose **Storage > EVS** to view the EVS disk. Alternatively, you can view the EVS disk based on the volume name on the EVS console.

---Fin

## 18.2.3 (kubectl) Creación de un PV a partir de un disco de EVS existente

### Notes and Constraints

The following configuration example applies to clusters of Kubernetes 1.13 or earlier.

### Procedimiento

**Paso 1** Inicie sesión en la consola de EVS, cree un disco de EVS y registre el ID de volumen, la capacidad y el tipo de disco del disco de EVS.

**Paso 2** Utilice kubectl para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).

**Paso 3** Cree dos archivos YAML para crear el PersistentVolume (PV) y el PersistentVolumeClaim (PVC). Suponga que los nombres de archivo son **pv-evs-example.yaml** y **pvc-evs-example.yaml**.

```
touch pv-evs-example.yaml pvc-evs-example.yaml
```

Versión de clúster de Kubernetes	Descripción	Ejemplo de YAML
1.11.7 ≤ versión de K8s ≤ 1.13	Clústeres de v1.11.7 a v1.13	<a href="#">Ejemplo de YAML</a>
1.11 ≤ versión de K8s < 1.11.7	Clústeres de v1.11 a v1.11.7	<a href="#">Ejemplo de YAML</a>
Versión de K8s = 1.9	Clústeres de v1.9	<a href="#">Ejemplo de YAML</a>

### Clústeres de v1.11.7 a v1.13

- **Ejemplo de archivo YAML para el PV:**

```

apiVersion: v1
kind: PersistentVolume
metadata:
  labels:
    failure-domain.beta.kubernetes.io/region: ap-southeast-1
    failure-domain.beta.kubernetes.io/zone: ap-southeast-1a
  annotations:
    pv.kubernetes.io/provisioned-by: flexvolume-huawei.com/fuxivol
  name: pv-evs-example
spec:
  accessModes:
    - ReadWriteOnce
  capacity:
    storage: 10Gi
  claimRef:
    apiVersion: v1
    kind: PersistentVolumeClaim
    name: pvc-evs-example
    namespace: default
  flexVolume:
    driver: huawei.com/fuxivol
    fsType: ext4
    options:
      disk-mode: SCSI
      fsType: ext4
      volumeID: 0992dbda-6340-470e-a74e-4f0db288ed82
    persistentVolumeReclaimPolicy: Delete
    storageClassName: sas
    
```

**Tabla 18-3** Parámetros de clave

Parámetro	Descripción
failure-domain.beta.kubernetes.io/region	La región donde se encuentra el clúster.
failure-domain.beta.kubernetes.io/zone	La AZ donde se crea el volumen de EVS. Debe ser la misma que la AZ prevista para la carga de trabajo.
storage	Capacidad de volumen de EVS en la unidad de Gi.
storageClassName	Tipo del disco de EVS. Valores admitidos: High I/O (SAS) and Ultra-high I/O (SSD)

Parámetro	Descripción
driver	Controlador de almacenamiento. Para los discos de EVS, establezca este parámetro en <a href="https://huawei.com/fuxivol">huawei.com/fuxivol</a> .
volumeID	ID de volumen del disco de EVS. Para obtener el ID de volumen, inicie sesión en la consola de CCE, elija <b>Resource Management</b> > <b>Storage</b> , haga clic en el nombre de PVC en la página de ficha <b>EVS</b> y copie el ID de PVC en la página de detalles de PVC.
disk-mode	Tipo de dispositivo del disco de EVS. El valor es <b>VBD</b> o <b>SCSI</b> . Para clústeres de CCE anteriores a v1.11.7, no es necesario establecer este campo. El valor predeterminado es <b>VBD</b> . Este campo es obligatorio para los clústeres de CCE de v1.11.7 a v1.13 que usan Linux x86. Como los volúmenes de EVS aprovisionados dinámicamente por un PVC se crean a partir de discos SCSI de EVS, se recomienda elegir <b>SCSI</b> al crear volúmenes manualmente (PV estáticos). Los volúmenes en el modo VBD todavía se pueden utilizar después de las actualizaciones del clúster.
spec.claimRef.apiVersion	El valor se fija en <b>v1</b> .
spec.claimRef.kind	El valor se fija en <b>PersistentVolumeClaim</b> .
spec.claimRef.name	Nombre de PVC. El valor es el mismo que el nombre del PVC creado en el siguiente paso.
spec.claimRef.namespace	Espacio de nombres del PVC. El valor es el mismo que el espacio de nombres del PVC creado en el siguiente paso.

● **Ejemplo de archivo YAML para el PVC:**

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    volume.beta.kubernetes.io/storage-class: sas
    volume.beta.kubernetes.io/storage-provisioner: flexvolume-huawei.com/fuxivol
  labels:
    failure-domain.beta.kubernetes.io/region: ap-southeast-1
    failure-domain.beta.kubernetes.io/zone: ap-southeast-1a
    name: pvc-evs-example
    namespace: default
spec:
  accessModes:
    - ReadWriteOnce
```

```
resources:
  requests:
    storage: 10Gi
  volumeName: pv-evs-example
```

**Tabla 18-4** Parámetros de clave

Parámetro	Descripción
volume.beta.kubernetes.io/storage-class	Clase de almacenamiento, que debe ser la misma que la del PV existente.
volume.beta.kubernetes.io/storage-provisioner	El campo debe estar establecido en <b>flexvolume-huawei.com/fuxivol</b> .
failure-domain.beta.kubernetes.io/region	La región donde se encuentra el clúster.
failure-domain.beta.kubernetes.io/zone	La AZ donde se crea el volumen de EVS. Debe ser la misma que la AZ prevista para la carga de trabajo.
storage	Capacidad solicitada en el PVC, en Gi. El valor debe ser el mismo que el tamaño de almacenamiento del PV existente.
volumeName	Nombre del PV.

**Clústeres de v1.11 a v1.11.7**

- **Ejemplo de archivo YAML para el PV:**

```
apiVersion: v1
kind: PersistentVolume
metadata:
  labels:
    failure-domain.beta.kubernetes.io/region: ap-southeast-1
    failure-domain.beta.kubernetes.io/zone: ap-southeast-1a
  name: pv-evs-example
spec:
  accessModes:
    - ReadWriteOnce
  capacity:
    storage: 10Gi
  flexVolume:
    driver: huawei.com/fuxivol
    fsType: ext4
    options:
      fsType: ext4
      volumeID: 0992dbda-6340-470e-a74e-4f0db288ed82
  persistentVolumeReclaimPolicy: Delete
  storageClassName: sas
```

**Tabla 18-5** Parámetros de clave

Parámetro	Descripción
failure-domain.beta.kubernetes.io/region	La región donde se encuentra el clúster.

Parámetro	Descripción
failure-domain.beta.kubernetes.io/zone	La AZ donde se crea el volumen de EVS. Debe ser la misma que la AZ prevista para la carga de trabajo.
storage	Capacidad de volumen de EVS en la unidad de Gi.
storageClassName	Tipo del disco de EVS. Valores admitidos: High I/O (SAS) and Ultra-high I/O (SSD)
driver	Controlador de almacenamiento. Para los discos de EVS, establezca este parámetro en <b>huawei.com/fuxivol</b> .
volumeID	ID de volumen del disco de EVS. Para obtener el ID de volumen, inicie sesión en la consola de CCE, elija <b>Resource Management &gt; Storage</b> , haga clic en el nombre de PVC en la página de ficha <b>EVS</b> y copie el ID de PVC en la página de detalles de PVC.
disk-mode	Tipo de dispositivo del disco de EVS. El valor es <b>VBD</b> o <b>SCSI</b> . Para clústeres de CCE anteriores a v1.11.7, no es necesario establecer este campo. El valor predeterminado es <b>VBD</b> . Este campo es obligatorio para los clústeres de CCE de v1.11.7 a v1.13 que usan Linux x86. Como los volúmenes de EVS aprovisionados dinámicamente por un PVC se crean a partir de discos SCSI de EVS, se recomienda elegir <b>SCSI</b> al crear volúmenes manualmente (PV estáticos). Los volúmenes en el modo VBD todavía se pueden utilizar después de las actualizaciones del clúster.

● **Ejemplo de archivo YAML para el PVC:**

```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    volume.beta.kubernetes.io/storage-class: sas
    volume.beta.kubernetes.io/storage-provisioner: flexvolume-huawei.com/fuxivol
  labels:
    failure-domain.beta.kubernetes.io/region: ap-southeast-1
    failure-domain.beta.kubernetes.io/zone: ap-southeast-1a
  name: pvc-evs-example
  namespace: default
spec:
  accessModes:
  - ReadWriteOnce
  resources:

```

```
requests:
  storage: 10Gi
volumeName: pv-evs-example
```

**Tabla 18-6** Parámetros de clave

Parámetro	Descripción
volume.beta.kubernetes.io/storage-class	Clase de almacenamiento. El valor puede ser <b>sas</b> o <b>ssd</b> . El valor debe ser el mismo que el del PV existente.
volume.beta.kubernetes.io/storage-provisioner	El campo debe estar establecido en <b>flexvolume-huawei.com/fuxivol</b> .
failure-domain.beta.kubernetes.io/region	La región donde se encuentra el clúster.
failure-domain.beta.kubernetes.io/zone	La AZ donde se crea el volumen de EVS. Debe ser la misma que la AZ prevista para la carga de trabajo.
storage	Capacidad solicitada en el PVC, en Gi. El valor debe ser el mismo que el tamaño de almacenamiento del PV existente.
volumeName	Nombre del PV.

### Clústeres de v1.9

- **Ejemplo de archivo YAML para el PV:**

```
apiVersion: v1
kind: PersistentVolume
metadata:
  labels:
    failure-domain.beta.kubernetes.io/region: ap-southeast-1
    failure-domain.beta.kubernetes.io/zone: ap-southeast-1a
  name: pv-evs-example
  namespace: default
spec:
  accessModes:
    - ReadWriteOnce
  capacity:
    storage: 10Gi
  flexVolume:
    driver: huawei.com/fuxivol
    fsType: ext4
    options:
      fsType: ext4
      kubernetes.io/namespace: default
      volumeID: 0992dbda-6340-470e-a74e-4f0db288ed82
  persistentVolumeReclaimPolicy: Delete
  storageClassName: sas
```

**Tabla 18-7** Parámetros de clave

Parámetro	Descripción
failure-domain.beta.kubernetes.io/region	La región donde se encuentra el clúster.
failure-domain.beta.kubernetes.io/zone	La AZ donde se crea el volumen de EVS. Debe ser la misma que la AZ prevista para la carga de trabajo.
storage	Capacidad de volumen de EVS en la unidad de Gi.
storageClassName	Tipo del disco de EVS. Valores admitidos: High I/O (SAS) and Ultra-high I/O (SSD)
driver	Controlador de almacenamiento. Para los discos de EVS, establezca este parámetro en <b>huawei.com/fuxivol</b> .
volumeID	ID de volumen del disco de EVS. Para obtener el ID de volumen, inicie sesión en la consola de CCE, elija <b>Resource Management &gt; Storage</b> , haga clic en el nombre de PVC en la página de ficha <b>EVS</b> y copie el ID de PVC en la página de detalles de PVC.
disk-mode	Tipo de dispositivo del disco de EVS. El valor es <b>VBD</b> o <b>SCSI</b> . Para clústeres de CCE anteriores a v1.11.7, no es necesario establecer este campo. El valor predeterminado es <b>VBD</b> . Este campo es obligatorio para los clústeres de CCE de v1.11.7 a v1.13 que usan Linux x86. Como los volúmenes de EVS aprovisionados dinámicamente por un PVC se crean a partir de discos SCSI de EVS, se recomienda elegir <b>SCSI</b> al crear volúmenes manualmente (PV estáticos). Los volúmenes en el modo VBD todavía se pueden utilizar después de las actualizaciones del clúster.

- **Ejemplo de archivo YAML para el PVC:**

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    volume.beta.kubernetes.io/storage-class: sas
    volume.beta.kubernetes.io/storage-provisioner: flexvolume-huawei.com/fuxivol
  labels:
    failure-domain.beta.kubernetes.io/region: ap-southeast-1
    failure-domain.beta.kubernetes.io/zone: ap-southeast-1a
name: pvc-evs-example
```

```

namespace: default
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
  volumeName: pv-evs-example
  volumeNamespace: default
    
```

**Tabla 18-8** Parámetros de clave

Parámetro	Descripción
volume.beta.kubernetes.io/storage-class	Clase de almacenamiento, que debe ser la misma que la del PV existente.
volume.beta.kubernetes.io/storage-provisioner	El campo debe estar establecido en <b>flexvolume-huawei.com/fuxivol</b> .
failure-domain.beta.kubernetes.io/region	La región donde se encuentra el clúster.
failure-domain.beta.kubernetes.io/zone	La AZ donde se crea el volumen de EVS. Debe ser la misma que la AZ prevista para la carga de trabajo.
storage	Capacidad solicitada en el PVC, en Gi. El valor debe ser el mismo que el tamaño de almacenamiento del PV existente.
volumeName	Nombre del PV.

**Paso 4** Cree un PV.

```
kubectl create -f pv-evs-example.yaml
```

**Paso 5** Cree un PVC.

```
kubectl create -f pvc-evs-example.yaml
```

Una vez que la operación se realice correctamente, elija **Resource Management > Storage** para ver el PVC creado. También puede ver el disco de EVS por su nombre en la consola de EVS.

**Paso 6** (Opcional) Agregue los metadatos asociados con el clúster para asegurarse de que el disco de EVS asociado con el PV estático montado no se elimina cuando se elimina el nodo o el clúster.

 **ATENCIÓN**

Si se omite este paso en este ejemplo o al crear un PV o PVC estático, asegúrese de que el disco de EVS asociado con el PV estático no está vinculado del nodo antes de eliminar el nodo.

1. Obtain the tenant token. For details, see [Obtaining a User Token](#).

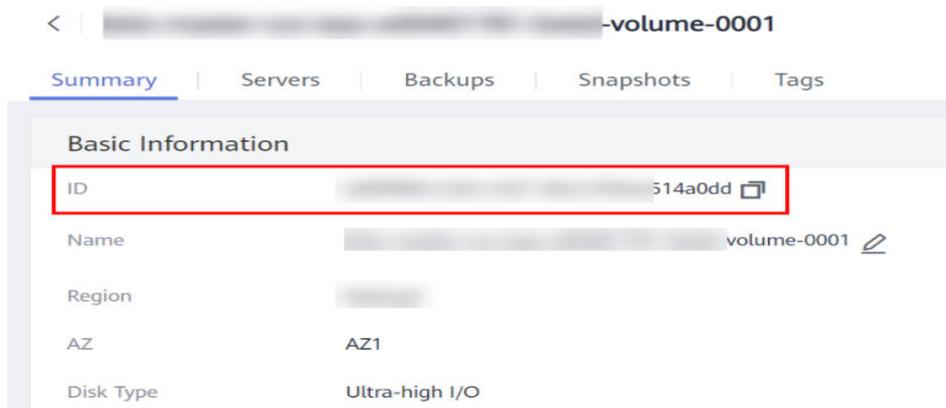
2. Obtain the EVS access address **EVS\_ENDPOINT**. For details, see [Regions and Endpoints](#).
3. Agregue los metadatos asociados con el clúster al disco de EVS que respalda el PV estático.

```
curl -X POST ${EVS_ENDPOINT}/v2/${project_id}/volumes/${volume_id}/metadata --insecure \
-d '{"metadata":{"cluster_id": "${cluster_id}", "namespace": "${pvc_namespace}"}}' \
-H 'Accept:application/json' -H 'Content-Type:application/json; charset=utf8' \
-H 'X-Auth-Token:${TOKEN}'
```

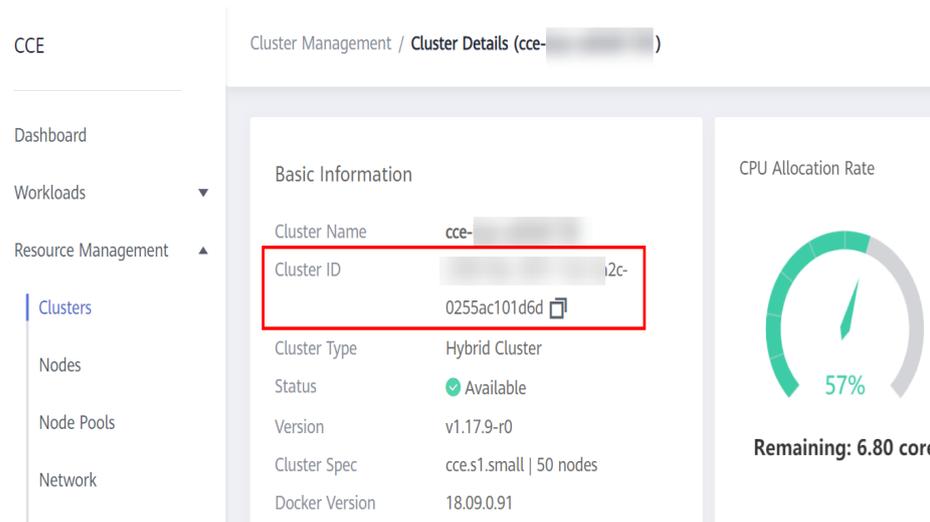
**Tabla 18-9** Parámetros de clave

Parámetro	Descripción
EVS_ENDPOINT	Dirección de acceso de EVS. Establezca este parámetro en el valor obtenido en <a href="#">Paso 6.2</a> .
project_id	ID del proyecto.
volume_id	ID del disco de EVS asociado. Establezca este parámetro en <b>volume_id</b> del PV estático que se va a crear. También puede iniciar sesión en la consola de EVS, hacer clic en el nombre del disco de EVS que se va a importar y obtener el ID de <b>Summary</b> en la página de detalles del disco como se muestra en <a href="#">Figura 18-2</a> .
cluster_id	ID del clúster donde se va a crear el PV de EVS. En la consola de CCE, elija <b>Resource Management &gt; Clusters</b> . Haga clic en el nombre del clúster que se va a asociar. En la página de detalles del clúster, obtenga el identificador del clúster, tal como se muestra en <a href="#">Figura 18-3</a> .
pvc_namespace	Espacio de nombres donde se debe vincular el PVC.
TOKEN	Token de usuario. Establezca este parámetro en el valor obtenido en <a href="#">Paso 6.1</a> .

**Figura 18-2** Obtención del ID de disco



**Figura 18-3** Obtención del ID de clúster



Por ejemplo, ejecute los siguientes comandos:

```
curl -X POST https://evs.ap-southeast-1.myhuaweicloud.com:443/v2/060576866680d5762f52c0150e726aa7/volumes/69c9619d-174c-4c41-837e-31b892604e14/metadata --insecure \
-d '{"metadata":{"cluster_id": "71e8277e-80c7-11ea-925c-0255ac100442",
"namespace": "default"}}' \
-H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' \
-H 'X-Auth-Token:MIIPe*****IsIm1ldG
```

Después de ejecutar la solicitud, ejecute los siguientes comandos para comprobar si el disco de EVS se ha asociado con los metadatos del clúster:

```
curl -X GET ${EVS_ENDPOINT}/v2/${project_id}/volumes/${volume_id}/metadata --insecure \
-H 'X-Auth-Token:${TOKEN}'
```

Por ejemplo, ejecute los siguientes comandos:

```
curl -X GET https://evs.ap-southeast-1.myhuaweicloud.com/v2/060576866680d5762f52c0150e726aa7/volumes/69c9619d-174c-4c41-837e-31b892604e14/metadata --insecure \
-H 'X-Auth-Token:MIIPeAYJ***9t1c31ASaQ=='
```

La salida del comando muestra los metadatos actuales del disco de EVS.

```
{
  "metadata": {
    "namespace": "default",
    "cluster_id": "71e8277e-80c7-11ea-925c-0255ac100442",
    "hw:passthrough": "true"
  }
}
```

----Fin

## 18.2.4 (kubectl) Creating a Pod Mounted with an EVS Volume

### Scenario

After an EVS volume is created or imported to CCE, you can mount it to a workload.

**AVISO**

EVS disks cannot be attached across AZs. Before mounting a volume, you can run the **kubectl get pvc** command to query the available PVCs in the AZ where the current cluster is located.

**Notes and Constraints**

The following configuration example applies to clusters of Kubernetes 1.13 or earlier.

**Procedure**

**Paso 1** Use kubectl to connect to the cluster. For details, see [Conexión a un clúster con kubectl](#).

**Paso 2** Run the following commands to configure the **evs-deployment-example.yaml** file, which is used to create a Deployment.

**touch evs-deployment-example.yaml**

**vi evs-deployment-example.yaml**

Example of mounting an EVS volume to a Deployment (PVC-based, shared volume):

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: evs-deployment-example
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      app: evs-deployment-example
  template:
    metadata:
      labels:
        app: evs-deployment-example
    spec:
      containers:
        - image: nginx
          name: container-0
          volumeMounts:
            - mountPath: /tmp
              name: pvc-evs-example
      imagePullSecrets:
        - name: default-secret
      restartPolicy: Always
      volumes:
        - name: pvc-evs-example
          persistentVolumeClaim:
            claimName: pvc-evs-auto-example
```

**Tabla 18-10** Key parameters

Parent Parameter	Parameter	Description
spec.template.spec.containers.volumeMounts	name	Name of the volume mounted to the container.

Parent Parameter	Parameter	Description
spec.template.spec.containers.volumeMounts	mountPath	Mount path of the container. In this example, the volume is mounted to the <b>/tmp</b> directory.
spec.template.spec.volumes	name	Name of the volume.
spec.template.spec.volumes.persistentVolumeClaim	claimName	Name of an existing PVC.

 **NOTA**

**spec.template.spec.containers.volumeMounts.name** and **spec.template.spec.volumes.name** must be consistent because they have a mapping relationship.

Mounting an EVS volume to a StatefulSet (PVC template-based, non-shared volume):

**Example YAML:**

```

apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: deploy-evs-sas-in
spec:
  replicas: 1
  selector:
    matchLabels:
      app: deploy-evs-sata-in
  template:
    metadata:
      labels:
        app: deploy-evs-sata-in
        failure-domain.beta.kubernetes.io/region: ap-southeast-1
        failure-domain.beta.kubernetes.io/zone: ap-southeast-1a
    spec:
      containers:
        - name: container-0
          image: 'nginx:1.12-alpine-perl'
          volumeMounts:
            - name: bs-sas-mountoptionpvc
              mountPath: /tmp
          imagePullSecrets:
            - name: default-secret
      volumeClaimTemplates:
        - metadata:
            name: bs-sas-mountoptionpvc
            annotations:
              volume.beta.kubernetes.io/storage-class: sas
              volume.beta.kubernetes.io/storage-provisioner: flexvolume-huawei.com/
  fuxivol
    spec:
      accessModes:
        - ReadWriteOnce
      resources:
        requests:
          storage: 10Gi
      serviceName: www
    
```

**Tabla 18-11** Key parameters

Parent Parameter	Parameter	Description
metadata	name	Name of the created workload.
spec.template.spec.containers	image	Image of the workload.
spec.template.spec.containers.volumeMount	mountPath	Mount path of the container. In this example, the volume is mounted to the <b>/tmp</b> directory.
spec	serviceName	Service corresponding to the workload. For details about how to create a Service, see <a href="#">Creación de un StatefulSet</a> .

 **NOTA**

`spec.template.spec.containers.volumeMounts.name` and `spec.volumeClaimTemplates.metadata.name` must be consistent because they have a mapping relationship.

**Paso 3** Run the following command to create the pod:

```
kubectl create -f evs-deployment-example.yaml
```

After the creation is complete, log in to the CCE console. In the navigation pane, choose **Resource Management > Storage > EVS**. Then, click the PVC name. On the PVC details page, you can view the binding relationship between the EVS volume and the PVC.

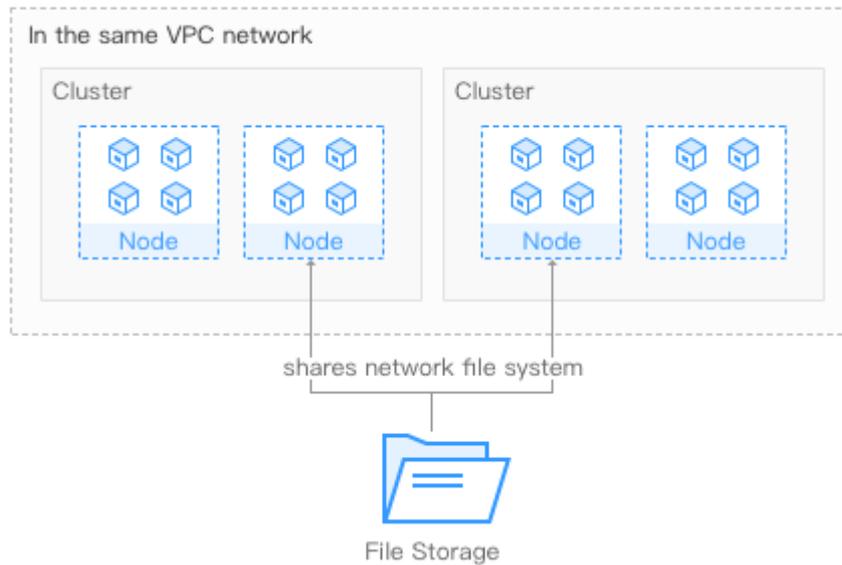
---Fin

## 18.3 Using SFS Turbo File Systems as Storage Volumes

### 18.3.1 Overview

CCE allows you to mount a volume created from an SFS Turbo file system to a container to store data persistently. Provisioned on demand and fast, SFS Turbo is suitable for DevOps, container microservices, and enterprise OA scenarios.

**Figura 18-4** Mounting SFS Turbo volumes to CCE



## Description

- **Standard file protocols:** You can mount file systems as volumes to servers, the same as using local directories.
- **Data sharing:** The same file system can be mounted to multiple servers, so that data can be shared.
- **Private network:** User can access data only in private networks of data centers.
- **Data isolation:** The on-cloud storage service provides exclusive cloud file storage, which delivers data isolation and ensures IOPS performance.
- **Use cases:** Deployments/StatefulSets in the ReadWriteMany mode, DaemonSets, and jobs created for high-traffic websites, log storage, DevOps, and enterprise OA applications

## 18.3.2 (kubectl) Creating a PV from an Existing SFS Turbo File System

### Scenario

CCE allows you to use an existing SFS Turbo file system to create a PersistentVolume (PV). After the creation is successful, you can create a PersistentVolumeClaim (PVC) and bind it to the PV.

### Notes and Constraints

The following configuration example applies to clusters of Kubernetes 1.13 or earlier.

### Procedure

- Paso 1** Log in to the SFS console, create a file system, and record the file system ID, shared path, and capacity.

**Paso 2** Use kubectl to connect to the cluster. For details, see [Conexión a un clúster con kubectl](#).

**Paso 3** Create two YAML files for creating the PV and PVC. Assume that the file names are **pv-efs-example.yaml** and **pvc-efs-example.yaml**.

**touch pv-efs-example.yaml pvc-efs-example.yaml**

- **Example YAML file for the PV:**

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-efs-example
  annotations:
    pv.kubernetes.io/provisioned-by: flexvolume-huawei.com/fuxiefs
spec:
  accessModes:
    - ReadWriteMany
  capacity:
    storage: 100Gi
  claimRef:
    apiVersion: v1
    kind: PersistentVolumeClaim
    name: pvc-efs-example
    namespace: default
  flexVolume:
    driver: huawei.com/fuxiefs
    fsType: efs
    options:
      deviceMountPath: <your_deviceMountPath> # Shared storage path of your
      SFS Turbo file.
      fsType: efs
      volumeID: 8962a2a2-a583-4b7f-bb74-fe76712d8414
    persistentVolumeReclaimPolicy: Delete
    storageClassName: efs-standard
```

**Tabla 18-12** Key parameters

Parameter	Description
driver	Storage driver used to mount the volume. Set it to <b>huawei.com/fuxiefs</b> .
deviceMountPath	Shared path of the SFS Turbo volume.
volumeID	SFS Turbo volume ID. To obtain the ID, log in to the CCE console, choose <b>Resource Management &gt; Storage</b> , click the PVC name in the <b>SFS Turbo</b> tab page, and copy the PVC ID on the PVC details page.
storage	File system size.
storageClassName	Volume type supported by SFS Turbo. The value can be <b>efs-standard</b> and <b>efs-performance</b> . Currently, SFS Turbo does not support dynamic creation; therefore, this parameter is not used for now.
spec.claimRef.apiVersion	The value is fixed at <b>v1</b> .
spec.claimRef.kind	The value is fixed at <b>PersistentVolumeClaim</b> .

Parameter	Description
spec.claimRef.name	The value is the same as the name of the PVC created in the next step.
spec.claimRef.namespace	The value is the same as the namespace of the PVC created in the next step.

● **Example YAML file for the PVC:**

```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    volume.beta.kubernetes.io/storage-class: efs-standard
    volume.beta.kubernetes.io/storage-provisioner: flexvolume-huawei.com/
    fuxiefs
  name: pvc-efs-example
  namespace: default
spec:
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 100Gi
    volumeName: pv-efs-example
    
```

**Tabla 18-13** Key parameters

Parameter	Description
volume.beta.kubernetes.io/ storage-class	Read/write mode supported by SFS Turbo. The value can be <b>efs-standard</b> or <b>efs-performance</b> . The value must be the same as that of the existing PV.
volume.beta.kubernetes.io/ storage-provisioner	The field must be set to <b>flexvolume-huawei.com/fuxiefs</b> .
storage	Storage capacity, in the unit of Gi. The value must be the same as the storage size of the existing PV.
volumeName	Name of the PV.

 **NOTA**

The VPC to which the SFS Turbo file system belongs must be the same as the VPC of the ECS VM planned for the workload. Ports 111, 445, 2049, 2051, and 20048 must be enabled in the security groups.

**Paso 4** Create the PV.

**kubectl create -f pv-efs-example.yaml**

**Paso 5** Create the PVC.

**kubectl create -f pvc-efs-example.yaml**

**----Fin**

## 18.3.3 (kubectl) Creating a Deployment Mounted with an SFS Turbo Volume

### Scenario

After an SFS Turbo volume is created or imported to CCE, you can mount the volume to a workload.

### Notes and Constraints

The following configuration example applies to clusters of Kubernetes 1.13 or earlier.

### Procedure

**Paso 1** Use kubectl to connect to the cluster. For details, see [Conexión a un clúster con kubectl](#).

**Paso 2** Run the following commands to configure the `efs-deployment-example.yaml` file, which is used to create a Deployment:

```
touch efs-deployment-example.yaml
```

```
vi efs-deployment-example.yaml
```

Example of mounting an SFS Turbo volume to a Deployment (PVC-based, shared volume):

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: efs-deployment-example           # Workload name
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      app: efs-deployment-example
  template:
    metadata:
      labels:
        app: efs-deployment-example
    spec:
      containers:
        - image: nginx
          name: container-0
          volumeMounts:
            - mountPath: /tmp           # Mount path
              name: pvc-efs-example
      restartPolicy: Always
      imagePullSecrets:
        - name: default-secret
      volumes:
        - name: pvc-efs-example
          persistentVolumeClaim:
            claimName: pvc-sfs-auto-example           # PVC name
```

**Tabla 18-14** Key parameters

Parameter	Description
name	Name of the created Deployment.

Parameter	Description
app	Name of the application running in the Deployment.
mountPath	Mount path in the container. In this example, the mount path is / <b>tmp</b> .

 **NOTA**

`spec.template.spec.containers.volumeMounts.name` and `spec.template.spec.volumes.name` must be consistent because they have a mapping relationship.

**Paso 3** Run the following command to create the pod:

```
kubectl create -f efs-deployment-example.yaml
```

After the creation is complete, choose **Storage > SFS Turbo** on the CCE console and click the PVC name. On the PVC details page, you can view the binding relationship between SFS Turbo and PVC.

----Fin

## 18.3.4 (kubectl) Creating a StatefulSet Mounted with an SFS Turbo Volume

### Scenario

CCE allows you to use an existing SFS Turbo volume to create a StatefulSet.

### Notes and Constraints

The following configuration example applies to clusters of Kubernetes 1.13 or earlier.

### Procedure

**Paso 1** Create an SFS Turbo volume and record the volume name.

**Paso 2** Use kubectl to connect to the cluster. For details, see [Conexión a un clúster con kubectl](#).

**Paso 3** Create a YAML file for creating the workload. Assume that the file name is `efs-statefulset-example.yaml`.

```
touch efs-statefulset-example.yaml
```

```
vi efs-statefulset-example.yaml
```

**Example YAML:**

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: efs-statefulset-example
  namespace: default
spec:
  replicas: 1
  selector:
```

```

matchLabels:
  app: efs-statefulset-example
template:
  metadata:
    annotations:
      metrics.alpha.kubernetes.io/custom-endpoints:
' [{"api":"","path":"","port":"","names":""}] '
      pod.alpha.kubernetes.io/initialized: 'true'
    labels:
      app: efs-statefulset-example
  spec:
    containers:
      - image: 'nginx:1.0.0'
        name: container-0
        resources:
          requests: {}
          limits: {}
        env:
          - name: PAAS_APP_NAME
            value: efs-statefulset-example
          - name: PAAS_NAMESPACE
            value: default
          - name: PAAS_PROJECT_ID
            value: b18296881cc34f929baa8b9e95abf88b
        volumeMounts:
          - name: efs-statefulset-example
            mountPath: /tmp
            readOnly: false
            subPath: ''
    imagePullSecrets:
      - name: default-secret
    terminationGracePeriodSeconds: 30
    volumes:
      - persistentVolumeClaim:
          claimName: cce-efs-import-jnr48lgm-3y5o
          name: efs-statefulset-example
    affinity: {}
    tolerations:
      - key: node.kubernetes.io/not-ready
        operator: Exists
        effect: NoExecute
        tolerationSeconds: 300
      - key: node.kubernetes.io/unreachable
        operator: Exists
        effect: NoExecute
        tolerationSeconds: 300
    podManagementPolicy: OrderedReady
    serviceName: test
    updateStrategy:
      type: RollingUpdate
    
```

**Tabla 18-15** Key parameters

Parameter	Description
replicas	Number of pods.
name	Name of the created workload.
image	Image used by the workload.
mountPath	Mount path in the container.
serviceName	Service corresponding to the workload. For details about how to create a Service, see <a href="#">Creación de un StatefulSet</a> .

Parameter	Description
claimName	Name of an existing PVC.

**NOTA**

`spec.template.spec.containers.volumeMounts.name` and `spec.template.spec.volumes.name` must be consistent because they have a mapping relationship.

**Paso 4** Create the StatefulSet.

```
kubectl create -f efs-statefulset-example.yaml
```

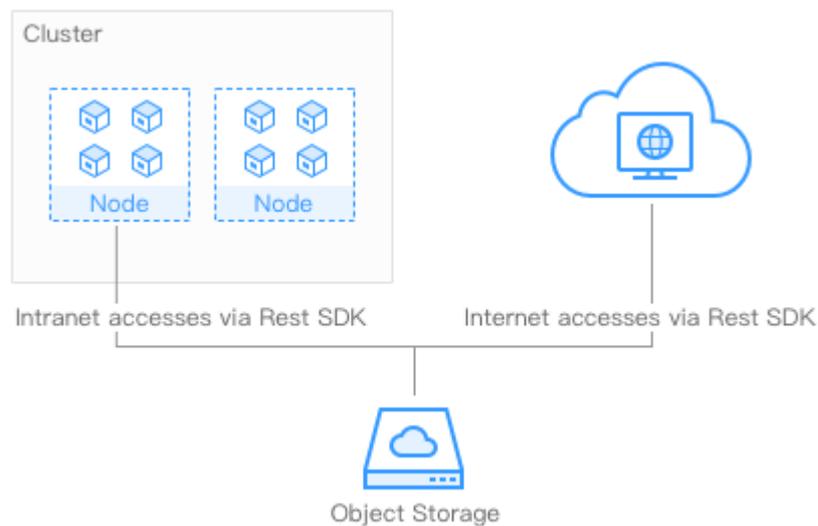
---Fin

## 18.4 Using OBS Buckets as Storage Volumes

### 18.4.1 Overview

CCE allows you to mount a volume created from an Object Storage Service (OBS) bucket to a container to store data persistently. Object storage is commonly used in cloud workloads, data analysis, content analysis, and hotspot objects.

**Figura 18-5** Mounting OBS volumes to CCE



### Notes and Constraints

Secure containers do not support OBS volumes.

A single user can create a maximum of 100 OBS buckets on the console. If you have a large number of CCE workloads and you want to mount an OBS bucket to every workload, you may easily run out of buckets. In this scenario, you are advised to use OBS through the OBS API or SDK and do not mount OBS buckets to the workload on the console.

## Storage Class

Object storage offers three storage classes, Standard, Infrequent Access, and Archive, to satisfy different requirements for storage performance and costs.

- The Standard storage class features low access latency and high throughput. It is therefore applicable to storing a large number of hot files (frequently accessed every month) or small files (less than 1 MB). The application scenarios include big data analytics, mobile apps, hot videos, and picture processing on social media.
- The Infrequent Access storage class is ideal for storing data that is semi-frequently accessed (less than 12 times a year), with requirements for quick response. The application scenarios include file synchronization or sharing, and enterprise-level backup. It provides the same durability, access latency, and throughput as the Standard storage class but at a lower cost. However, the Infrequent Access storage class has lower availability than the Standard storage class.
- The Archive storage class is suitable for archiving data that is rarely-accessed (averagely once a year). The application scenarios include data archiving and long-term data backup. The Archive storage class is secure and durable at an affordable low cost, which can be used to replace tape libraries. However, it may take hours to restore data from the Archive storage class.

## Description

- **Standard APIs:** With HTTP RESTful APIs, OBS allows you to use client tools or third-party tools to access object storage.
- **Data sharing:** Servers, embedded devices, and IoT devices can use the same path to access shared object data in OBS.
- **Public/Private networks:** OBS allows data to be accessed from public networks to meet Internet application requirements.
- **Capacity and performance:** No capacity limit; high performance (read/write I/O latency within 10 ms).
- **Use cases:** Deployments/StatefulSets in the ReadOnlyMany mode and jobs created for big data analysis, static website hosting, online video on demand (VoD), gene sequencing, intelligent video surveillance, backup and archiving, and enterprise cloud boxes (web disks). You can create object storage by using the OBS console, tools, and SDKs.

## Reference

CCE clusters can also be mounted with OBS buckets of third-party tenants, including OBS parallel file systems (preferred) and OBS object buckets. For details, see [Mounting an Object Storage Bucket of a Third-Party Tenant](#).

## 18.4.2 (kubectl) Automatically Creating an OBS Volume

### Scenario

During the use of OBS, expected OBS buckets can be automatically created and mounted as volumes. Currently, standard and infrequent access OBS buckets are supported, which correspond to **obs-standard** and **obs-standard-ia**, respectively.

## Notes and Constraints

The following configuration example applies to clusters of Kubernetes 1.13 or earlier.

## Procedure

**Paso 1** Use `kubectl` to connect to the cluster. For details, see [Conexión a un clúster con kubectl](#).

**Paso 2** Run the following commands to configure the `pvc-obs-auto-example.yaml` file, which is used to create a PVC.

```
touch pvc-obs-auto-example.yaml
```

```
vi pvc-obs-auto-example.yaml
```

### Example YAML:

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    volume.beta.kubernetes.io/storage-class: obs-standard # OBS bucket type. The
value can be obs-standard (standard) or obs-standard-ia (infrequent access).
  name: pvc-obs-auto-example # PVC name
  namespace: default
spec:
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 1Gi # Storage capacity in the unit of Gi. For OBS buckets, this
parameter is used only for verification (fixed to 1, cannot be empty or 0). Any
value you set does not take effect for OBS buckets.
```

**Tabla 18-16** Key parameters

Parameter	Description
volume.beta.kubernetes.io/storage-class	Bucket type. Currently, <b>obs-standard</b> and <b>obs-standard-ia</b> are supported.
name	Name of the PVC to be created.
accessModes	Only <b>ReadWriteMany</b> is supported. <b>ReadWriteOnly</b> is not supported.
storage	Storage capacity in the unit of Gi. For OBS buckets, this field is used only for verification (cannot be empty or 0). Its value is fixed at <b>1</b> , and any value you set does not take effect for OBS buckets.

**Paso 3** Run the following command to create a PVC:

```
kubectl create -f pvc-obs-auto-example.yaml
```

After the command is executed, an OBS bucket is created in the VPC to which the cluster belongs. You can click the bucket name in **Storage > OBS** to view the bucket or view it on the OBS console.

----Fin

## 18.4.3 (kubectl) Creación de un PV a partir de un bucket de OBS existente

### Escenario

CCE le permite usar un bucket de OBS existente para crear un PersistentVolume (PV). Puede crear un PersistentVolumeClaim (PVC) y vincularlo al PV.

### Notes and Constraints

The following configuration example applies to clusters of Kubernetes 1.13 or earlier.

### Procedimiento

- Paso 1** Inicie sesión en la consola de OBS, cree un bucket de OBS y registre el nombre del bucket y la clase de almacenamiento.
- Paso 2** Utilice kubectl para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).
- Paso 3** Cree dos archivos YAML para crear el PV y el PVC. Suponga que los nombres de archivo son **pv-obs-example.yaml** y **pvc-obs-example.yaml**.

**touch pv-obs-example.yaml pvc-obs-example.yaml**

Versión de clúster de Kubernetes	Descripción	Ejemplo de YAML
1.11 ≤ versión de K8s ≤ 1.13	Clústeres de v1.11 a v1.13	<a href="#">Ejemplo de YAML</a>
Versión de K8s = 1.9	Clústeres de v1.9	<a href="#">Ejemplo de YAML</a>

### Clústeres desde v1.11 hasta v1.13

- **Ejemplo de archivo YAML para el PV:**

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-obs-example
  annotations:
    pv.kubernetes.io/provisioned-by: flexvolume-huawei.com/fuxiobs
spec:
  accessModes:
  - ReadWriteMany
  capacity:
    storage: 1Gi
  claimRef:
    apiVersion: v1
    kind: PersistentVolumeClaim
    name: pvc-obs-example
    namespace: default
  flexVolume:
    driver: huawei.com/fuxiobs
    fsType: obs
    options:
      fsType: obs
```

```

region: ap-southeast-1
storage_class: STANDARD
volumeID: test-obs
persistentVolumeReclaimPolicy: Delete
storageClassName: obs-standard
    
```

**Tabla 18-17** Parámetros de clave

Parámetro	Descripción
driver	Controlador de almacenamiento utilizado para montar el volumen. Establezca el controlador en <a href="http://huawei.com/fuxiobs">huawei.com/fuxiobs</a> para el volumen de OBS.
storage_class	Clase de almacenamiento, incluidos <b>STANDARD</b> (bucket estándar) y <b>STANDARD_IA</b> (bucket de acceso poco frecuente).
region	La región donde se encuentra el clúster.
volumeID	Nombre del bucket de OBS. Para obtener el nombre, inicie sesión en la consola de CCE, elija <b>Resource Management &gt; Storage</b> , haga clic en el nombre de PVC en la página de fichas <b>OBS</b> y copie el nombre de PV en la página de fichas <b>PV Details</b> .
storage	Capacidad de almacenamiento, en Gi. El valor se fija en <b>1Gi</b> .
storageClassName	Clase de almacenamiento compatible con OBS, incluidos <b>obs-standard</b> (bucket estándar) y <b>obs-standard-ia</b> (bucket de acceso poco frecuente).
spec.claimRef.apiVersion	El valor se fija en <b>v1</b> .
spec.claimRef.kind	El valor se fija en <b>PersistentVolumeClaim</b> .
spec.claimRef.name	El valor es el mismo que el nombre del PVC creado en el siguiente paso.
spec.claimRef.nameSpace	El valor es el mismo que el espacio de nombres del PVC creado en el siguiente paso.

- **Ejemplo de archivo YAML para el PVC:**

```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    volume.beta.kubernetes.io/storage-class: obs-standard
    volume.beta.kubernetes.io/storage-provisioner: flexvolume-huawei.com/fuxiobs
  name: pvc-obs-example
  namespace: default
spec:
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  volumeName: pv-obs-example
    
```

**Tabla 18-18** Parámetros de clave

Parámetro	Descripción
volume.beta.kubernetes.io/ storage-class	Clase de almacenamiento compatible con OBS, incluidos <b>obs-standard</b> y <b>obs-standard-ia</b> .
volume.beta.kubernetes.io/ storage-provisioner	Se debe establecer en <b>flexvolume-huawei.com/fuxiobs</b> .
volumeName	Nombre del PV.
storage	Capacidad de almacenamiento, en Gi. El valor se fija en <b>1Gi</b> .

### Clústeres de v1.9

- **Ejemplo de archivo YAML para el PV:**

```

apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-obs-example
  namespace: default
spec:
  accessModes:
    - ReadWriteMany
  capacity:
    storage: 1Gi
  flexVolume:
    driver: huawei.com/fuxiobs
    fsType: obs
    options:
      fsType: obs
      kubernetes.io/namespace: default
      region: ap-southeast-1
      storage_class: STANDARD
      volumeID: test-obs
  persistentVolumeReclaimPolicy: Delete
  storageClassName: obs-standard
    
```

**Tabla 18-19** Parámetros de clave

Parámetro	Descripción
driver	Controlador de almacenamiento utilizado para montar el volumen. Establezca el controlador en <b>huawei.com/fuxiobs</b> para el volumen de OBS.
storage_class	Clase de almacenamiento, incluidos <b>STANDARD</b> (bucket estándar) y <b>STANDARD_IA</b> (bucket de acceso poco frecuente).
region	La región donde se encuentra el clúster.
volumeID	Nombre del bucket de OBS.  Para obtener el nombre, inicie sesión en la consola de CCE, elija <b>Resource Management &gt; Storage</b> , haga clic en el nombre de PVC en la página de fichas <b>OBS</b> y copie el nombre de PV en la página de fichas <b>PV Details</b> .

Parámetro	Descripción
storage	Capacidad de almacenamiento, en Gi. El valor se fija en <b>1Gi</b> .
storageClassName	Clase de almacenamiento compatible con OBS, incluidos <b>obs-standard</b> (bucket estándar) y <b>obs-standard-ia</b> (bucket de acceso poco frecuente).

● **Ejemplo de archivo YAML para el PVC:**

```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    volume.beta.kubernetes.io/storage-class: obs-standard
    volume.beta.kubernetes.io/storage-provisioner: flexvolume-huawei.com/
    fuxiobs
  name: pvc-obs-example
  namespace: default
spec:
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  volumeName: pv-obs-example
  volumeNamespace: default
    
```

**Tabla 18-20** Parámetros de clave

Parámetro	Descripción
volume.beta.kubernetes.io/ storage-class	Clase de almacenamiento compatible con OBS, incluidos <b>obs-standard</b> y <b>obs-standard-ia</b> .
volume.beta.kubernetes.io/ storage-provisioner	Se debe establecer en <b>flexvolume-huawei.com/fuxiobs</b> .
volumeName	Nombre del PV.
storage	Capacidad de almacenamiento, en Gi. El valor se fija en <b>1Gi</b> .

**Paso 4** Cree un PV.

```
kubectl create -f pv-obs-example.yaml
```

**Paso 5** Cree un PVC.

```
kubectl create -f pvc-obs-example.yaml
```

----Fin

## 18.4.4 (kubectl) Creating a Deployment Mounted with an OBS Volume

### Scenario

After an OBS volume is created or imported to CCE, you can mount the volume to a workload.

### Notes and Constraints

The following configuration example applies to clusters of Kubernetes 1.13 or earlier.

### Procedure

**Paso 1** Use kubectl to connect to the cluster. For details, see [Conexión a un clúster con kubectl](#).

**Paso 2** Run the following commands to configure the **obs-deployment-example.yaml** file, which is used to create a pod.

```
touch obs-deployment-example.yaml
```

```
vi obs-deployment-example.yaml
```

Example of mounting an OBS volume to a Deployment (PVC-based, shared volume):

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: obs-deployment-example           # Workload name
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      app: obs-deployment-example
  template:
    metadata:
      labels:
        app: obs-deployment-example
    spec:
      containers:
        - image: nginx
          name: container-0
          volumeMounts:
            - mountPath: /tmp           # Mount path
              name: pvc-obs-example
      restartPolicy: Always
      imagePullSecrets:
        - name: default-secret
      volumes:
        - name: pvc-obs-example
          persistentVolumeClaim:
            claimName: pvc-obs-auto-example   # PVC name
```

**Tabla 18-21** Key parameters

Parameter	Description
name	Name of the pod to be created.
app	Name of the application running in the pod.

Parameter	Description
mountPath	Mount path in the container.

 **NOTA**

**spec.template.spec.containers.volumeMounts.name** and **spec.template.spec.volumes.name** must be consistent because they have a mapping relationship.

Example of mounting an OBS volume to a StatefulSet (PVC template-based, dedicated volume):

**Example YAML:**

```

apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: deploy-obs-standard-in
  namespace: default
  generation: 1
  labels:
    appgroup: ''
spec:
  replicas: 1
  selector:
    matchLabels:
      app: deploy-obs-standard-in
  template:
    metadata:
      labels:
        app: deploy-obs-standard-in
      annotations:
        metrics.alpha.kubernetes.io/custom-endpoints:
' [{"api":"","path":"","port":"","names":""}]'
        pod.alpha.kubernetes.io/initialized: 'true'
    spec:
      containers:
        - name: container-0
          image: 'nginx:1.12-alpine-perl'
          env:
            - name: PAAS_APP_NAME
              value: deploy-obs-standard-in
            - name: PAAS_NAMESPACE
              value: default
            - name: PAAS_PROJECT_ID
              value: a2cd8e998dca42e98a41f596c636dbda
          resources: {}
          volumeMounts:
            - name: obs-bs-standard-mountoptionpvc
              mountPath: /tmp
          terminationMessagePath: /dev/termination-log
          terminationMessagePolicy: File
          imagePullPolicy: IfNotPresent
      restartPolicy: Always
      terminationGracePeriodSeconds: 30
      dnsPolicy: ClusterFirst
      securityContext: {}
      imagePullSecrets:
        - name: default-secret
      affinity: {}
      schedulerName: default-scheduler
  volumeClaimTemplates:
    - metadata:
        name: obs-bs-standard-mountoptionpvc
        annotations:

```

```

        volume.beta.kubernetes.io/storage-class: obs-standard
        volume.beta.kubernetes.io/storage-provisioner: flexvolume-huawei.com/
fuxiobs
  spec:
    accessModes:
      - ReadWriteMany
    resources:
      requests:
        storage: 1Gi
    serviceName: www
    podManagementPolicy: OrderedReady
    updateStrategy:
      type: RollingUpdate
    revisionHistoryLimit: 10
    
```

**Tabla 18-22** Key parameters

Parameter	Description
name	Name of the created workload.
image	Image of the workload.
mountPath	Mount path in the container. In this example, the volume is mounted to the <b>/tmp</b> directory.
serviceName	Service corresponding to the workload. For details about how to create a Service, see <a href="#">Creación de un StatefulSet</a> .

 **NOTA**

**spec.template.spec.containers.volumeMounts.name** and **spec.volumeClaimTemplates.metadata.name** must be consistent because they have a mapping relationship.

**Paso 3** Run the following command to create the pod:

```
kubectl create -f obs-deployment-example.yaml
```

After the creation is complete, choose **Storage > OBS** on the CCE console and click the PVC name. On the PVC details page, you can view the binding relationship between the OBS service and the PVC.

----Fin

## 18.4.5 (kubectl) Creating a StatefulSet Mounted with an OBS Volume

### Scenario

CCE allows you to use an existing OBS volume to create a StatefulSet through a PersistentVolumeClaim (PVC).

### Notes and Constraints

The following configuration example applies to clusters of Kubernetes 1.13 or earlier.

## Procedure

- Paso 1** Create an OBS volume by referring to [\(kubectl\) Automatically Creating an OBS Volume](#) and obtain the PVC name.
- Paso 2** Use kubectl to connect to the cluster. For details, see [Conexión a un clúster con kubectl](#).
- Paso 3** Create a YAML file for creating the workload. Assume that the file name is **obs-statefulset-example.yaml**.

**touch obs-statefulset-example.yaml**

**vi obs-statefulset-example.yaml**

### Example YAML:

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: obs-statefulset-example
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      app: obs-statefulset-example
  serviceName: qwqq
  template:
    metadata:
      annotations:
        metrics.alpha.kubernetes.io/custom-endpoints:
        '[{"api":"","path":"","port":"","names":""}]'
        pod.alpha.kubernetes.io/initialized: "true"
      creationTimestamp: null
      labels:
        app: obs-statefulset-example
    spec:
      affinity: {}
      containers:
        image: nginx:latest
        imagePullPolicy: Always
        name: container-0
        volumeMounts:
        - mountPath: /tmp
          name: pvc-obs-example
      imagePullSecrets:
        - name: default-secret
      volumes:
        - name: pvc-obs-example
          persistentVolumeClaim:
            claimName: cce-obs-demo
```

**Tabla 18-23** Key parameters

Parameter	Description
replicas	Number of pods.
name	Name of the created workload.
image	Image used by the workload.
mountPath	Mount path in the container.

Parameter	Description
serviceName	Service corresponding to the workload. For details about how to create a Service, see <a href="#">Creación de un StatefulSet</a> .
claimName	Name of an existing PVC.

**Paso 4** Create the StatefulSet.

```
kubectl create -f obs-statefulset-example.yaml
```

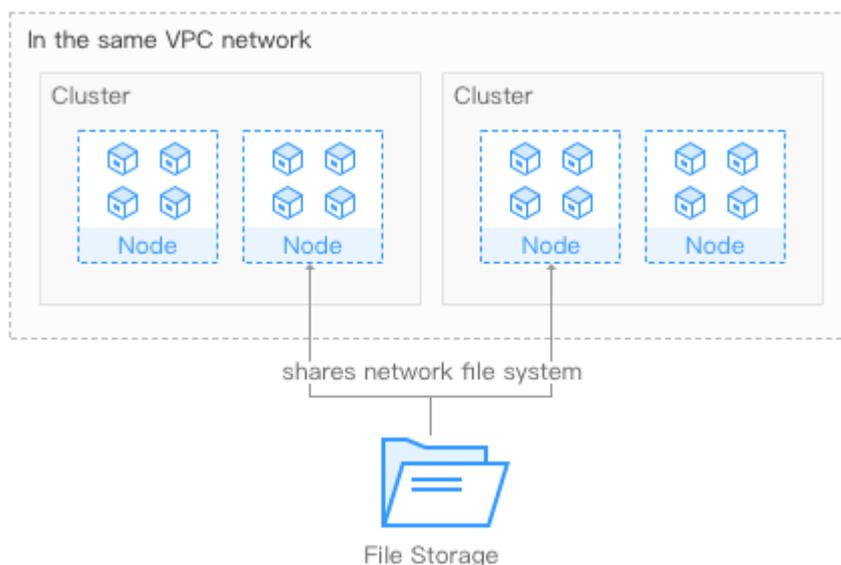
----Fin

## 18.5 Using SFS File Systems as Storage Volumes

### 18.5.1 Overview

CCE allows you to mount a volume created from a Scalable File Service (SFS) file system to a container to store data persistently. SFS volumes are commonly used in ReadWriteMany scenarios, such as media processing, content management, big data analysis, and workload process analysis.

**Figura 18-6** Mounting SFS volumes to CCE



### Description

- **Standard file protocols:** You can mount file systems as volumes to servers, the same as using local directories.
- **Data sharing:** The same file system can be mounted to multiple servers, so that data can be shared.
- **Private network:** User can access data only in private networks of data centers.

- **Capacity and performance:** The capacity of a single file system is high (PB level) and the performance is excellent (ms-level read/write I/O latency).
- **Use cases:** Deployments/StatefulSets in the ReadWriteMany mode and jobs created for high-performance computing (HPC), media processing, content management, web services, big data analysis, and workload process analysis

For details, see [SFS Service Overview](#).

## 18.5.2 (kubectl) Automatically Creating an SFS Volume

### **NOTA**

Currently, SFS file systems are sold out and PVCs cannot be automatically created using the storage class.

### Notes and Constraints

The following configuration example applies to clusters of Kubernetes 1.13 or earlier.

### Procedure

**Paso 1** Use kubectl to connect to the cluster. For details, see [Conexión a un clúster con kubectl](#).

**Paso 2** Run the following commands to configure the `pvc-sfs-auto-example.yaml` file, which is used to create a PVC.

```
touch pvc-sfs-auto-example.yaml
```

```
vi pvc-sfs-auto-example.yaml
```

#### Example YAML file:

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    volume.beta.kubernetes.io/storage-class: nfs-rw
  name: pvc-sfs-auto-example
  namespace: default
spec:
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
```

**Tabla 18-24** Key parameters

Parameter	Description
volume.beta.kubernetes.io/storage-class	File storage class. Currently, the standard file protocol type (nfs-rw) is supported.
name	Name of the PVC to be created.
accessModes	Only <b>ReadWriteMany</b> is supported. <b>ReadWriteOnly</b> is not supported.
storage	Storage capacity in the unit of Gi.

**Paso 3** Run the following command to create a PVC:

```
kubectl create -f pvc-sfs-auto-example.yaml
```

After the command is executed, a file system is created in the VPC to which the cluster belongs. Choose **Storage > SFS** on the CCE console or log in to the SFS console to view the file system.

----Fin

## 18.5.3 (kubectl) Creación de un PV a partir de un sistema de archivos de SFS existente

### Notes and Constraints

The following configuration example applies to clusters of Kubernetes 1.13 or earlier.

### Procedimiento

- Paso 1** Inicie sesión en la consola de SFS, cree un sistema de archivos y registre el ID del sistema de archivos, la ruta compartida y la capacidad.
- Paso 2** Utilice kubectl para conectarse al clúster. Para obtener más información, véase [Conexión a un clúster con kubectl](#).
- Paso 3** Cree dos archivos YAML para crear el PV y el PVC. Suponga que los nombres de archivo son **pv-sfs-example.yaml** y **pvc-sfs-example.yaml**.

```
touch pv-sfs-example.yaml pvc-sfs-example.yaml
```

Versión de clúster de Kubernetes	Descripción	Ejemplo de YAML
1.11 ≤ la versión de K8s < 1.13	Clústeres de v1.11 a v1.13	<a href="#">Ejemplo de YAML</a>
Versión de K8s = 1.9	Clústeres de v1.9	<a href="#">Ejemplo de YAML</a>

### Clústeres desde v1.11 a v1.13

- **Ejemplo de archivo YAML para el PV:**

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-sfs-example
  annotations:
    pv.kubernetes.io/provisioned-by: flexvolume-huawei.com/fuxinfs
spec:
  accessModes:
    - ReadWriteMany
  capacity:
    storage: 10Gi
  claimRef:
    apiVersion: v1
    kind: PersistentVolumeClaim
    name: pvc-sfs-example
    namespace: default
```

```
flexVolume:
  driver: huawei.com/fuxinfs
  fsType: nfs
  options:
    deviceMountPath: <your_deviceMountPath> # Shared storage path of your
file.
    fsType: nfs
    volumeID: f6976f9e-2493-419b-97ca-d7816008d91c
  persistentVolumeReclaimPolicy: Delete
  storageClassName: nfs-rw
```

**Tabla 18-25** Parámetros de clave

Parámetro	Descripción
driver	Controlador de almacenamiento utilizado para montar el volumen. Establezca el controlador en <b>huawei.com/fuxinfs</b> para el sistema de archivos.
deviceMountPath	Ruta de acceso compartida del sistema de archivos. En la consola de gestión, elija <b>Service List &gt;Storage &gt;Scalable File Service</b> . Puede obtener la ruta compartida del sistema de archivos desde la columna <b>Mount Address</b> , como se muestra en <b>Figura 18-7</b> .
volumeID	ID del sistema de archivos. Para obtener el ID, inicie sesión en la consola de CCE, elija <b>Resource Management &gt; Storage</b> , haga clic en el nombre de PVC en la página de ficha <b>SFS</b> y copie el ID de PVC en la página de detalles de PVC.
storage	Tamaño del sistema de archivos.
storageClassName	Modo de lectura/escritura compatible con el sistema de archivos. Actualmente, son compatibles con <b>nfs-rw</b> y <b>nfs-ro</b> .
spec.claimRef.apiVersion	El valor se fija en <b>v1</b> .
spec.claimRef.kind	El valor se fija en <b>PersistentVolumeClaim</b> .
spec.claimRef.name	El valor es el mismo que el nombre del PVC creado en el siguiente paso.
spec.claimRef.nameSpace	Espacio de nombres del PVC. El valor es el mismo que el espacio de nombres del PVC creado en el siguiente paso.

- **Ejemplo de archivo YAML para el PVC:**

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    volume.beta.kubernetes.io/storage-class: nfs-rw
    volume.beta.kubernetes.io/storage-provisioner: flexvolume-huawei.com/
fuxinfs
  name: pvc-sfs-example
  namespace: default
spec:
  accessModes:
    - ReadWriteMany
```

```
resources:
  requests:
    storage: 10Gi
  volumeName: pv-sfs-example
```

**Tabla 18-26** Parámetros de clave

Parámetro	Descripción
volume.beta.kubernetes.io/ storage-class	Modo de lectura/escritura compatible con el sistema de archivos. <b>nfs-rw</b> y <b>nfs-ro</b> son compatibles. El valor debe ser el mismo que el del PV existente.
volume.beta.kubernetes.io/ storage-provisioner	Se debe establecer en <b>flexvolume-huawei.com/fuxinfs</b> .
storage	Capacidad de almacenamiento, en la unidad de Gi. El valor debe ser el mismo que el tamaño de almacenamiento del PV existente.
volumeName	Nombre del PV.

### Clústeres de v1.9

- **Ejemplo de archivo YAML para el PV:**

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-sfs-example
  namespace: default
spec:
  accessModes:
    - ReadWriteMany
  capacity:
    storage: 10Gi
  flexVolume:
    driver: huawei.com/fuxinfs
    fsType: nfs
    options:
      deviceMountPath: <your_deviceMountPath> # Shared storage path of your
file.
      fsType: nfs
      kubernetes.io/namespace: default
      volumeID: f6976f9e-2493-419b-97ca-d7816008d91c
  persistentVolumeReclaimPolicy: Delete
  storageClassName: nfs-rw
```

**Tabla 18-27** Parámetros de clave

Parámetro	Descripción
driver	Controlador de almacenamiento utilizado para montar el volumen. Establezca el controlador en <b>huawei.com/fuxinfs</b> para el sistema de archivos.

Parámetro	Descripción
deviceMountPath	Ruta de acceso compartida del sistema de archivos. En la consola de gestión, elija <b>Service List &gt;Storage &gt;Scalable File Service</b> . Puede obtener la ruta compartida del sistema de archivos desde la columna <b>Mount Address</b> , como se muestra en <b>Figura 18-7</b> .
volumeID	ID del sistema de archivos. Para obtener el ID, inicie sesión en la consola de CCE, elija <b>Resource Management &gt; Storage</b> , haga clic en el nombre de PVC en la página de ficha <b>SFS</b> y copie el ID de PVC en la página de detalles de PVC.
storage	Tamaño del sistema de archivos.
storageClassName	Modo de lectura/escritura compatible con el sistema de archivos. Actualmente, son compatibles con <b>nfs-rw</b> y <b>nfs-ro</b> .

● **Ejemplo de archivo YAML para el PVC:**

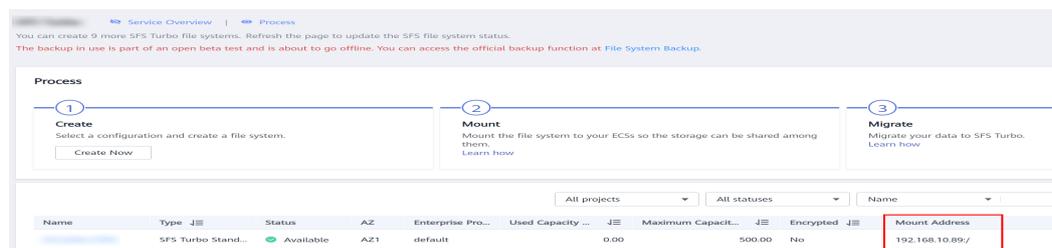
```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    volume.beta.kubernetes.io/storage-class: nfs-rw
    volume.beta.kubernetes.io/storage-provisioner: flexvolume-huawei.com/
  fuxinfs
  name: pvc-sfs-example
  namespace: default
spec:
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
    volumeName: pv-sfs-example
    volumeNamespace: default
    
```

**Tabla 18-28** Parámetros de clave

Parámetro	Descripción
volume.beta.kubernetes.io/ storage-class	Modo de lectura/escritura compatible con el sistema de archivos. <b>nfs-rw</b> y <b>nfs-ro</b> son compatibles. El valor debe ser el mismo que el del PV existente.
volume.beta.kubernetes.io/ storage-provisioner	El campo debe estar establecido en <b>flexvolume-huawei.com/fuxinfs</b> .
storage	Capacidad de almacenamiento, en la unidad de Gi. El valor debe ser el mismo que el tamaño de almacenamiento del PV existente.
volumeName	Nombre del PV.

**Figura 18-7** SFS - dirección de montaje del sistema de archivos



**NOTA**

La VPC a la que pertenece el sistema de archivos debe ser la misma que la VPC de la máquina virtual de ECS para la que se planifica la carga de trabajo.

**Paso 4** Cree un PV.

```
kubectl create -f pv-sfs-example.yaml
```

**Paso 5** Cree un PVC.

```
kubectl create -f pvc-sfs-example.yaml
```

----Fin

## 18.5.4 (kubectl) Creating a Deployment Mounted with an SFS Volume

### Scenario

After an SFS volume is created or imported to CCE, you can mount the volume to a workload.

### Notes and Constraints

The following configuration example applies to clusters of Kubernetes 1.13 or earlier.

### Procedure

**Paso 1** Use `kubectl` to connect to the cluster. For details, see [Conexión a un clúster con kubectl](#).

**Paso 2** Run the following commands to configure the `sfs-deployment-example.yaml` file, which is used to create a pod.

```
touch sfs-deployment-example.yaml
```

```
vi sfs-deployment-example.yaml
```

Example of mounting an SFS volume to a Deployment (PVC-based, shared volume):

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: sfs-deployment-example # Workload name
  namespace: default
spec:
  replicas: 1
  selector:
```

```

matchLabels:
  app: sfs-deployment-example
template:
  metadata:
    labels:
      app: sfs-deployment-example
  spec:
    containers:
      - image: nginx
        name: container-0
        volumeMounts:
          - mountPath: /tmp # Mount path
            name: pvc-sfs-example
    imagePullSecrets:
      - name: default-secret
    restartPolicy: Always
    volumes:
      - name: pvc-sfs-example
        persistentVolumeClaim:
          claimName: pvc-sfs-auto-example # PVC name
    
```

**Tabla 18-29** Key parameters

Parent Parameter	Parameter	Description
metadata	name	Name of the pod to be created.
spec.template.spec.containers.volumeMounts	mountPath	Mount path in the container. In this example, the mount path is <b>/tmp</b> .
spec.template.spec.volumes.persistentVolumeClaim	claimName	Name of an existing PVC.

 **NOTA**

**spec.template.spec.containers.volumeMounts.name** and **spec.template.spec.volumes.name** must be consistent because they have a mapping relationship.

Example of mounting an SFS volume to a StatefulSet (PVC template-based, dedicated volume):

 **NOTA**

Currently, SFS file systems are sold out and cannot be exclusively used by defining the PVC template.

**Example YAML:**

```

apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: deploy-sfs-nfs-rw-in
  namespace: default
  labels:
    appgroup: ''
spec:
  replicas: 2
  selector:
    matchLabels:
      app: deploy-sfs-nfs-rw-in
  template:
    metadata:
      labels:
        app: deploy-sfs-nfs-rw-in
    
```

```
spec:
  containers:
    - name: container-0
      image: 'nginx:1.12-alpine-perl'
      volumeMounts:
        - name: bs-nfs-rw-mountoptionpvc
          mountPath: /aaa
      imagePullSecrets:
        - name: default-secret
  volumeClaimTemplates:
    - metadata:
        name: bs-nfs-rw-mountoptionpvc
        annotations:
          volume.beta.kubernetes.io/storage-class: nfs-rw
          volume.beta.kubernetes.io/storage-provisioner: flexvolume-huawei.com/
  fuxinfs
  spec:
    accessModes:
      - ReadWriteMany
    resources:
      requests:
        storage: 1Gi
    serviceName: www
```

**Tabla 18-30** Key parameters

Parent Parameter	Parameter	Description
metadata	name	Name of the created workload.
spec.template.spec.containers	image	Image of the workload.
spec.template.spec.containers.volumeMount	mountPath	Mount path in the container. In this example, the mount path is <b>/tmp</b> .
spec	serviceName	Service corresponding to the workload. For details about how to create a Service, see <a href="#">Creación de un StatefulSet</a> .

 **NOTA**

**spec.template.spec.containers.volumeMounts.name** and **spec.volumeClaimTemplates.metadata.name** must be consistent because they have a mapping relationship.

**Paso 3** Run the following command to create the pod:

**kubectl create -f sfs-deployment-example.yaml**

After the creation is complete, log in to the CCE console. In the navigation pane, choose **Resource Management > Storage > SFS**. Click the PVC name. On the PVC details page, you can view the binding relationship between SFS and PVC.

**----Fin**

## 18.5.5 (kubectl) Creating a StatefulSet Mounted with an SFS Volume

### Scenario

CCE allows you to use an existing SFS volume to create a StatefulSet through a PersistentVolumeClaim (PVC).

### Notes and Constraints

The following configuration example applies to clusters of Kubernetes 1.13 or earlier.

### Procedure

- Paso 1** Create an SFS volume by referring to [\(kubectl\) Automatically Creating an SFS Volume](#) and record the volume name.
- Paso 2** Use kubectl to connect to the cluster. For details, see [Conexión a un clúster con kubectl](#).
- Paso 3** Create a YAML file for creating the workload. Assume that the file name is **sfs-statefulset-example.yaml**.

```
touch sfs-statefulset-example.yaml
```

```
vi sfs-statefulset-example.yaml
```

#### Example YAML:

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: sfs-statefulset-example
  namespace: default
spec:
  replicas: 2
  selector:
    matchLabels:
      app: sfs-statefulset-example
  serviceName: qwqq
  template:
    metadata:
      annotations:
        metrics.alpha.kubernetes.io/custom-endpoints:
' [{"api": "", "path": "", "port": "", "names": ""}] '
        pod.alpha.kubernetes.io/initialized: "true"
      labels:
        app: sfs-statefulset-example
    spec:
      affinity: {}
      containers:
        - image: nginx:latest
          name: container-0
          volumeMounts:
            - mountPath: /tmp
              name: pvc-sfs-example
      imagePullSecrets:
        - name: default-secret
      volumes:
        - name: pvc-sfs-example
          persistentVolumeClaim:
            claimName: cce-sfs-demo
```

**Tabla 18-31** Key parameters

Parent Parameter	Parameter	Description
spec	replicas	Number of pods.
metadata	name	Name of the created workload.
spec.template.spec.containers	image	Image used by the workload.
spec.template.spec.containers. volumeMounts	mountPath	Mount path in the container.
spec	serviceName	Service corresponding to the workload. For details about how to create a Service, see <a href="#">Creación de un StatefulSet</a> .
spec.template.spec.volumes.persistentVolumeClaim	claimName	Name of an existing PVC.

 **NOTA**

`spec.template.spec.containers.volumeMounts.name` and `spec.template.spec.volumes.name` must be consistent because they have a mapping relationship.

**Paso 4** Create the StatefulSet.

**kubectl create -f sfs-statefulset-example .yaml**

**----Fin**