Cloud Container Engine Autopilot

Product Bulletin-In Use

Issue	01
Date	2025-01-16

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

Contents

1 Latest Notices

CCE Autopilot has released the latest notices.

No.	Title	Type	Release Date
1	Billing Changes for Huawei Cloud CCE Autopilot Data Plane	Product Change Notices	2024-09-14
2	CCE Autopilot for Commercial Use on September 30, 2024, 00:00 GMT+08:00	Product Change	2024-08-29
3	Notice of Kubernetes Security Vulnerability (CVE-2024-10220)	Vulnerability Notices	2024-12-04
4	Notice of the NGINX Ingress Controller Vulnerability That Allows Attackers to Bypass Annotation Validation (CVE-2024-7646)	Vulnerability Notices	2024-08-26
5	Notice of Fluent Bit Memory Corruption Vulnerability (CVE-2024-4323)	Vulnerability Notices	2024-05-23

For more historical notices, see **Product Change Notices** and **Vulnerability Notices**.

2 Product Change Notices

2.1 Billing Changes for Huawei Cloud CCE Autopilot Data Plane

Released: September 14, 2024

On September 18, 2024, 22:00:00 GMT+08:00, Huawei Cloud will change the CPU and memory billing for the CCE Autopilot data plane. This will result in a change of the product type in the Autopilot data plane resource bills from CCE to CCI. However, the unit prices and historical bills will remain the same, and your services will not be affected. The following are the details of the bill changes.

Table 2-1 Resource billing before adjustment

Product Type	Product	Billing Mode	Billed By	Price Unit	Specifications
Cloud Container Engine (CCE)	CCE Autopilot	Pay-per-use	Duration	USD per second	CCE memory
CCE	CCE Autopilot	Pay-per-use	Duration	USD per second	CCE CPU

Table 2-2 Resource billing after adjustment

Product Type	Product	Billing Mode	Billed By	Price Unit	Specifications
Cloud Container Instance (CCI)	CCI - Autopilot Resources	Pay-per-use	Duration	USD per second	Autopilot general-computing memory

Product Type	Product	Billing Mode	Billed By	Price Unit	Specifications
CCI	CCI - Autopilot Resources	Pay-per-use	Duration	USD per second	Autopilot general-computing CPU

If you have any questions or suggestions, **submit a service ticket**.

Thank you for using Huawei Cloud.

2.2 CCE Autopilot for Commercial Use on September 30, 2024, 00:00 GMT+08:00

Released: August 29, 2024

Huawei Cloud is set to launch CCE Autopilot for commercial use on September 30, 2024, 00:00 GMT+08:00.

Once the service is launched, you will be charged for cluster management, while other expenditures will remain the same as those during the OBT.

For more information about CCE Autopilot, see **What Is a CCE Autopilot Cluster?**

If you have any questions, **submit a service ticket**.

Thank you for using Huawei Cloud.

For details, see **Commercial Use Notice: CCE Autopilot Will Be Released Commercially on September 30, 2024, 00:00 GMT+08:00**.

3 Vulnerability Notices

3.1 Vulnerability Fixing Policies

Cluster Vulnerability Fixing SLA

- High-risk vulnerabilities:
  - CCE Autopilot fixes vulnerabilities within one month after the Kubernetes community detects them and releases fixing solutions. The fixing policies are the same as those of the community.
  - Emergency vulnerabilities of the operating system are released according to the operating system fixing policies and procedure. Generally, a fixing solution is provided within one month, and you need to fix the vulnerabilities by yourself.
- Other vulnerabilities:

  Other vulnerabilities can be fixed through a normal upgrade.

Statement

To prevent customers from being exposed to unexpected risks, CCE Autopilot does not provide other information about the vulnerability except the vulnerability background, details, technical analysis, affected functions/versions/scenarios, solutions, and reference information.

In addition, CCE Autopilot provides the same information for all customers to protect all customers equally. CCE Autopilot will not notify individual customers in advance.

CCE Autopilot does not develop or release exploitable intrusive code (or code for verification) using the vulnerabilities in the product.

3.2 Notice of Kubernetes Security Vulnerability (CVE-2024-10220)

The Kubernetes community recently discovered a security vulnerability (CVE-2024-10220). This vulnerability allows an attacker who has the necessary

permissions to create pods associated with gitRepo volumes to run arbitrary commands outside the containers. The attacker can exploit the hooks directory in the target Git repository to escape the containers and execute malicious commands.

Description

Table 3-1 Vulnerability details

Type	CVE-ID	Severity	Discovered
Container escape	CVE-2024-10220	High	2024-11-22

Impact

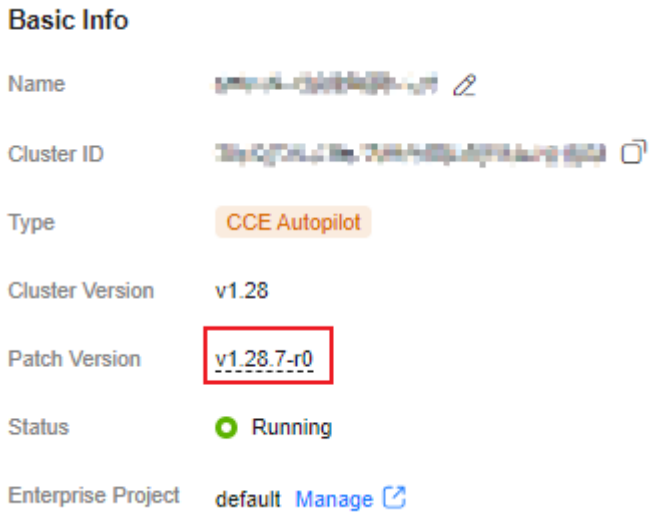
The affected cluster versions are as follows:

- v1.27.0-r0-v1.27.8-r0
- v1.28.0-r0-v1.28.6-r0

Identification Method

Log in to the CCE console, click the name of the target cluster to access the cluster console, and check the cluster version on the **Overview** page.

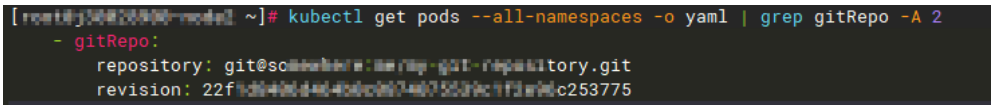
Figure 3-1 Cluster Version



- If the cluster version is not one of the versions mentioned above, then the vulnerability does not affect the cluster.
- If the cluster version falls within the affected range, you can use the following command to check if the vulnerability has been exploited in the cluster:

(This command will display a list of all gitRepo storage volumes that are mounted to pods. It will also clone the repository to the pod in the **.git** subdirectory.)

```
kubectl get pods --all-namespaces -o yaml | grep gitRepo -A 2
```



If the command output does not show any gitRepo configuration, it means that the cluster is not affected by the vulnerability.

Solution

- This vulnerability has been fixed for CCE Autopilot clusters. Upgrade the cluster to the version where the vulnerability has been fixed promptly. For clusters that have reached EOS, upgrade them to versions under maintenance.

  Fixed cluster versions: v1.27.9-r0, v1.28.7-r0, and later
- The **gitRepo storage volumes** are no longer supported. As a solution, the community recommends using the init containers to perform Git clone operations and then mount the directories to the pods. For details, see **the example in GitHub**.

Helpful Links

https://github.com/kubernetes/kubernetes/issues/128885

3.3 Notice of the NGINX Ingress Controller Vulnerability That Allows Attackers to Bypass Annotation Validation (CVE-2024-7646)

Description

Table 3-2 Vulnerability details

Type	CVE-ID	Severity	Discovered
Validation bypass and command injection	CVE-2024-7646	Critical	2024-08-19

Impact

Attackers with permissions to create ingresses in Kubernetes clusters (in networking.k8s.io or extensions API group) can exploit a vulnerability in ingress-nginx earlier than v1.11.2. This allows them to bypass annotation validation and inject arbitrary commands, potentially gaining access to the credentials of the ingress-nginx controller and sensitive information in a cluster.

This vulnerability is involved when the NGINX Ingress Controller add-on earlier than 2.4.14 is installed in a CCE Autopilot cluster.

Identification Method

1. Use kubectl to search for pods related to **cceaddon-nginx-ingress**.
   kubectl get po -A | grep cceaddon-nginx-ingress

   ```
   [root@192-168-53-14 paas]# kubectl get po -A|grep cceaddon-nginx-ingress
   kube-system   cceaddon-nginx-ingress-controller-67bff65f66-h8xlt        1/1     Running
   kube-system   cceaddon-nginx-ingress-default-backend-699d6f4578-nqqqr   1/1     Running
   ```

   If similar information is displayed, the NGINX Ingress Controller add-on has been installed in the cluster.
2. Check the nginx-ingress image version used by the NGINX Ingress Controller add-on.
   kubectl get deploy cceaddon-nginx-ingress-controller -nkube-system -oyaml|grep -w image

   ```
   [root@192-168-53-14 paas]# kubectl get deploy cceaddon-nginx-ingress-controller -nkube-system -oyaml|grep -w image
         image: [redacted]hwofficial/nginx-ingress:v1.11.2
         image: [redacted]hwofficial/nginx-ingress:v1.11.2
   ```

   If the installed NGINX Ingress Controller add-on has an nginx-ingress version earlier than v1.11.2, this vulnerability is present.

Solution

This vulnerability has been fixed for the NGINX Ingress Controller add-on in the CCE Autopilot cluster. Upgrade the add-on to the version where the vulnerability has been fixed.

Fixed add-on version: 2.4.14 or later

Helpful Links

Fixed version released by the community: **https://github.com/kubernetes/ingress-nginx/releases/tag/controller-v1.11.2**

3.4 Notice of Fluent Bit Memory Corruption Vulnerability (CVE-2024-4323)

Fluent Bit is a powerful, flexible, and user-friendly tool for processing and forwarding logs. It can be used with applications and systems of all sizes and types, including Linux, Windows, embedded Linux, and macOS. Fluent Bit is a widely used logging tool among cloud providers and enterprises, with over 13 billion downloads and deployments to date.

Description

Table 3-3 Vulnerability details

Type	CVE-ID	Severity	Discovered
Buffer overflow	CVE-2024-4323	Critical	2024-05-20

Impact

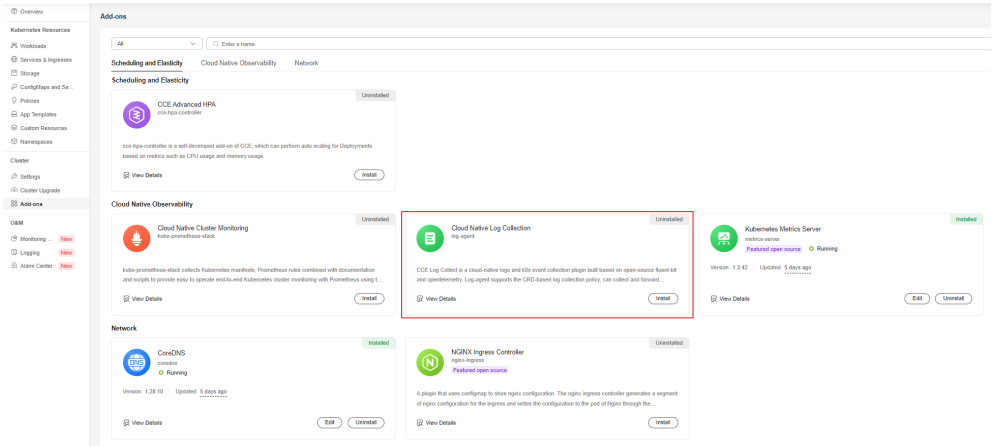
Fluent Bit 2.0.7 to 3.0.3 have a heap buffer overflow vulnerability in the embedded HTTP server's parsing of trace requests. The vulnerability arises from the incorrect verification of the data type of **input_name** during the parsing of incoming requests for the /api/v1/traces endpoint. This allows non-string values, including integer values, to be transferred in the inputs array of requests, which can lead to memory corruption. Attackers can exploit this vulnerability to cause a denial of service, information leakage, or remote code execution.

This vulnerability is involved when the Cloud Native Log Collection add-on earlier than 1.7.0 is installed in the CCE Autopilot cluster.

Identification Method

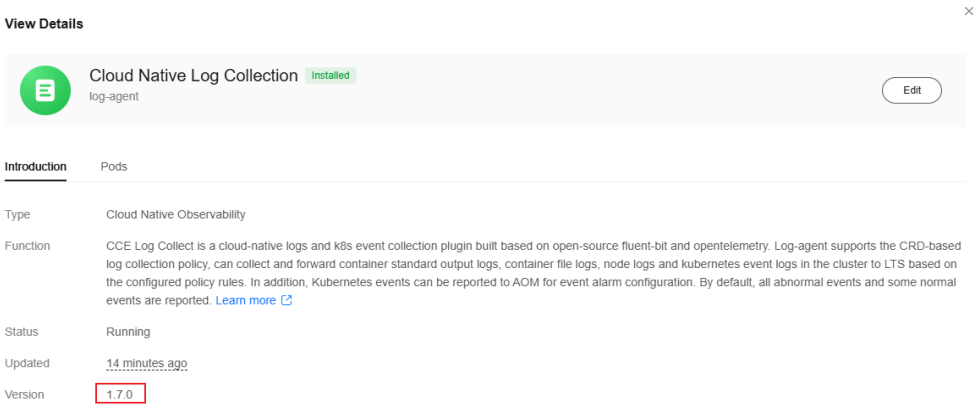
1. Go to the **Add-ons** page and check if the Cloud Native Log Collection add-in has been installed.

   Figure 3-2 Viewing the installed add-on version

2. In the Cloud Native Log Collection add-on details, view the add-on version. If the add-on version is earlier than 1.7.0, this vulnerability is involved.

   Figure 3-3 Add-on details

Solution

This vulnerability has been fixed for the Cloud Native Log Collection add-on in the CCE Autopilot cluster. Upgrade the add-on to the version where the vulnerability has been fixed.

Fixed add-on version: 1.7.0 or later

4 Product Release Notes

4.1 Cluster Versions

4.1.1 Kubernetes Version Policy

CCE provides highly scalable, high-performance, enterprise-class Kubernetes clusters. As the Kubernetes community periodically releases Kubernetes versions, CCE Autopilot will release Open Beta Test (OBT) and commercially used cluster versions accordingly. This section describes the Kubernetes version policy of CCE Autopilot clusters.

Lifecycle of CCE Autopilot Cluster Versions

Kubernetes Version	Status	Community Release In	OBT	Commercial Use	EOS of CCE Autopilot Clusters
v1.31	OBT	August 2024	December 2024	-	-
v1.28	In commercial use	August 2023	April 2024	September 2024	February 2026
v1.27	In commercial use	April 2023	February 2024	September 2024	October 2025

Phases of CCE Autopilot Cluster Versions

- OBT: You can experience the latest features of this cluster version. However, the stability of clusters of this version has not been completely verified, and the Service Level Agreement (SLA) of CCE Autopilot is not valid for such clusters.

- In commercial use: The cluster version has been fully verified and is stable and reliable. You can use clusters of this version in the production environment, and the CCE Autopilot SLA is valid for such clusters.
- EOS: After the cluster version EOS, CCE does not support the creation of new clusters or provide technical support including new feature updates, vulnerability or issue fixes, new patches, work order guidance, and online checks for the EOS cluster version. The CCE Autopilot SLA is not valid for such clusters.

CCE Autopilot Cluster Versions

CCE Autopilot clusters are updated according to the versions available in the Kubernetes community. This means that a CCE Autopilot cluster version is made up of both the Kubernetes community version number and the patch version number. The CCE cluster version is in the format of **v***X.Y.Z*-**r***N*, such as **v1.28.2-r0**.

- A Kubernetes version is in the format of *X.Y.Z*, which inherits the community version policy. The major Kubernetes version is represented by *X*, the minor Kubernetes version is represented by *Y*, and the Kubernetes patch version is represented by *Z*. For details, see **the Kubernetes version policies**. For details about the Kubernetes versions supported by CCE Autopilot, see **Kubernetes Version Release Notes**.
- A CCE Autopilot patch version is in the format of, for example, v1.30.4-r*N*. New patches are released on an irregular basis for Kubernetes versions that are still in the maintenance period. If a new patch version provides new features, bug fixes, vulnerability fixes, or scenario optimizations compared with the previous version, the *N* version number increases. For details about the patch versions, see **CCE Autopilot Cluster Patch Release History**.

Cluster Upgrade

Periodically upgrade CCE Autopilot clusters for better user experience. Using an EOS version, you cannot obtain technical support and CCE Autopilot SLA assurance. Upgrade CCE Autopilot clusters in a timely manner.

On the CCE console, you can easily upgrade clusters in a visualized manner, improving the stability and reliability of clusters. For details, see **Upgrade Overview**.

4.1.2 Kubernetes Version Release Notes

4.1.2.1 Kubernetes 1.31 Release Notes (OBT)

CCE Autopilot has passed the Certified Kubernetes Conformance Program and is a certified Kubernetes offering. CCE Autopilot allows you to create Kubernetes clusters 1.31. This topic describes the changes made in Kubernetes 1.31.

Indexes

- **New and Enhanced Features**
- **API Changes and Removals**

- **References**

New and Enhanced Features

Kubernetes 1.31

- Start ordinal of a StatefulSet

  StatefulSet start ordinal moved to the General Availability (GA) state in Kubernetes 1.31. By default, each pod in a StatefulSet is assigned an integer ordinal from 0. With this feature, you can configure a start ordinal for each pod. For details, see **Start ordinal**.

- Elastic indexed jobs

  Elastic indexed jobs moved to GA in Kubernetes 1.31. You can scale indexed Jobs up or down by modifying fields **.spec.completions** and **.spec.parallelism**. For details, see **Elastic Indexed Jobs**.

- Pod failure policy

  Pod failure policies moved to GA in Kubernetes 1.31. This feature helps you handle pod failures based on the container exit codes and pod conditions. For details, see **Pod failure policy**.

- Pod disruption conditions

  Pod disruption conditions moved to GA in Kubernetes 1.31. The new **DisruptionTarget** condition indicates that the pod is about to be deleted due to a disruption. The **reason** field indicates one of the following reasons for the pod termination: preempted by a pod with a higher priority, the pod has been cleared due to node deletion, or the pod is terminated by kubelet. When a pod is created using a job or CronJob, you can use these pod disruption conditions as part of your job's **pod failure policy** to define the action when a pod is abnormal. For details, see **Pod disruption conditions**.

- Selectable fields for custom resources

  Selectable fields for custom resources moved to Beta in Kubernetes 1.31. You can specify the **selectableFields** field of a **CustomResourceDefinition** to define which other fields in a custom resource may be used in field selectors. Field selectors can then be used to get only resources by filtering List, Watch, and DeleteCollection requests. For details, see **Selectable fields for custom resources**.

- Job success policy

  Job success policies moved to Beta in Kubernetes 1.31. When creating an indexed Job, you can define when a job can be declared as succeeded, based on the number of pods that succeeded. For details, see **Success policy**.

- matchLabelKeys

  matchLabelKeys moved to Beta in Kubernetes 1.31. matchLabelKeys and mismatchLabelKeys are finer fields for pod affinity or anti-affinity. They specify the keys for the labels that should or should not match with the incoming pod's labels, so that a rolling upgrade will not break affinity or anti-affinity. For details, see **matchLabelKeys**.

- ServiceAccountTokenNodeBinding

  ServiceAccountTokenNodeBinding moved to Beta in Kubernetes 1.31. You can create a service account token that is directly bound to a node. The token defines the node information and verifies whether the node is available. The

token will be valid until it expires or either the associated node is deleted. For details, see **Manually create an API token for a ServiceAccount**.

Kubernetes 1.30

- Webhook matching expression

  The Webhook matching expression feature moved to GA. This feature enables admission webhooks to be matched based on specific conditions, providing control over the triggering conditions of the webhooks in a more precise granularity. For details, see **Dynamic Admission Control**.

- Validating admission policies

  Validating admission policies moved to GA. This feature allows you to declare the validating admission policies of resources using Common Expression Language (CEL). For details, see **Validating Admission Policy**.

- Horizontal pod auto scaling based on container resource metrics

  The horizontal pod auto scaling feature based on container resource metrics advanced to GA. This feature allows HPA to configure auto scaling based on the resource usage of each container within a pod, rather than just the overall resource usage of the pod. This makes it easier to set scaling thresholds for the most critical containers in a pod. For details, see **Container resource metrics**.

- Legacy ServiceAccount token cleaner

  The legacy ServiceAccount token cleaner moved to GA. It runs as part of **kube-controller-manager** and checks every 24 hours to see if any auto-generated legacy ServiceAccount token has not been used in a specific amount of time (one year by default, specified by **--legacy-service-account-token-clean-up-period**). If so, the cleaner marks those tokens as invalid and adds the **kubernetes.io/legacy-token-invalid-since**, with the current date as the value. If an invalid token is not used for a specific period of time (one year by default, specified by **--legacy-service-account-token-clean-up-period**), the cleaner deletes it. For details, see **Legacy ServiceAccount token cleaner**.

Kubernetes 1.29

- Load balancer IP mode for Services

  The load balancer IP mode is a new alpha feature. Kubernetes 1.29 adds the **ipMode** field to the Services' **status** field for configuring traffic forwarding from Services within a cluster to pods. If **ipMode** is set to **VIP**, traffic to the load balancer will be redirected to the target node by kube-proxy. If it is set to **Proxy**, traffic delivered to a node will be sent to the load balancer and then redirected to the target node by the load balancer. This feature addresses the issue that the load balancer is not used to distribute traffic. For details, see **Load Balancer IP Mode for Services**.

- nftables proxy mode

  The nftables proxy mode is a new alpha feature. This feature allows kube-proxy to run in nftables mode. In this mode, kube-proxy configures packet forwarding rules using the nftables API of the kernel netfilter subsystem. For details, see **nftables proxy mode**.

- Garbage collection for unused container images

  The garbage collection for unused container images is a new alpha feature. This feature allows you to specify the maximum time a local image can be

unused for each node. If the time expires, the image will be garbage collected. To configure the setting, specify the **ImageMaximumGCAge** field for kubelet. For details, see **Garbage collection for unused container images**.

- **PodLifecycleSleepAction**

  PodLifecycleSleepAction is a new alpha feature. This feature introduces the sleep hook to the container lifecycle hooks. You can pause a container for a specified duration after it starts or before it is stopped by enabling this feature. For details, see **Hook handler implementations**.

- **KubeletSeparateDiskGC**

  KubeletSeparateDiskGC is a new alpha feature. With this feature enabled, container images and containers can be garbage collected even if they are on separate file systems.

- **matchLabelKeys** and **mismatchLabelKeys**

  matchLabelKeys and **mismatchLabelKeys** are new alpha features. With these features enabled, the **matchLabelKeys** and **mismatchLabelKeys** fields are added to the pod affinity and anti-affinity configurations. This allows for configurations of more affinity and anti-affinity policies between pods. For details, see **matchLabelKeys and mismatchLabelKeys**.

- **clusterTrustBundle** projected volumes

  clusterTrustBundle projected volumes are new alpha features. With this feature enabled, the **clusterTrustBundle** projected volume source injects the contents of one or more ClusterTrustBundle objects as an automatically-updating file. For details, see **clusterTrustBundle projected volumes**.

- Image pull per runtime class

  Image pull per runtime class is a new alpha feature. With this feature enabled, the kubelet references container images by a tuple (of image name or runtime handler) rather than just the image name or digest. Your container runtime may adapt its behavior based on the selected runtime handler. Pulling images based on runtime classes will be helpful for VM based containers. For details, see **Image pull per runtime class**.

- **PodReadyToStartContainers** condition

  The **PodReadyToStartContainers** moved to beta. Kubernetes 1.29 introduces the **PodReadyToStartContainers** condition to the pods' **status** field. If it is set to **true**, the sandbox of a pod is ready and service containers can be created. This feature enables cluster administrators to gain a clearer and more comprehensive view of pod sandbox creation completion and container readiness. This enhanced visibility allows them to make better-informed decisions and troubleshoot issues more effectively. For details, see **PodReadyToStartContainersCondition Moved to Beta**.

- Job-related features
  - Pod replacement policy

    The pod replacement policy feature moved to beta. This feature ensures that a pod is replaced only when it reaches the **Failed** state, which means that **status.phase** becomes **Failed**. It does not recreate a pod when the deletion timestamp is not empty and the pod is still being deleted. This prevents two pods from occupying index and node resources concurrently.
  - Backoff limit per index

    The backoff limit per index moved to beta. By default, pod failures for indexed jobs are counted and restricted by the global limit of retries,

specified by **.spec.backoffLimit**. This means that if there is a consistently failing index in a job, pods specified by the job will be restarted repeatedly until pod failures exhaust the limit. Once the limit is reached, the job is marked failed and pods for other indexes in the job may never be even started. The feature allows you to complete execution of all indexes, despite some indexes failing, and to better use the compute resources by avoiding unnecessary retries of consistently failing indexes.

- Native sidecar containers

  Native sidecar containers moved to beta. The **restartPolicy** field is added to **initContainers**. When this field is set to **Always**, the sidecar container is enabled. The sidecar container and service container are deployed in the same pod. This cannot prolong the pod lifecycle. Sidecar containers are commonly used in scenarios such as network proxy and log collection. For details, see **Sidecar Containers**.

- The legacy ServiceAccount token cleaner

  Legacy ServiceAccount token cleaner moved to beta. It runs as part of **kube-controller-manager** and checks every 24 hours to see if any auto-generated legacy ServiceAccount token has not been used in a specific amount of time (one year by default, specified by **--legacy-service-account-token-clean-up-period**). If so, the cleaner marks those tokens as invalid and adds the **kubernetes.io/legacy-token-invalid-since**, with the current date as the value. If an invalid token is not used for a specific period of time (one year by default, specified by **--legacy-service-account-token-clean-up-period**), the cleaner deletes it. For details, see **Legacy ServiceAccount Token Cleaner**.

- **DevicePluginCDIDevices**

  DevicePluginCDIDevices moved to beta. With this feature enabled, plugin developers can use the **CDIDevices** field added to **DeviceRunContainerOptions** to pass CDI device names directly to CDI enabled runtimes.

- **PodHostIPs**

  The **PodHostIPs** feature moved to beta. With this feature enabled, Kubernetes adds the **hostIPs** field to **Status** of pods and downward API to expose node IP addresses to workloads. This field specifies the dual-stack protocol version of the host IP address. The first IP address is always the same as the host IP address.

- API priority and fairness (APF)

  APF moved to GA. APF classifies and isolates requests in a more fine-grained way. It improves max-inflight limitations. It also introduces a limited amount of queuing, so that the API server does not reject any request in cases of very brief bursts. Requests are dispatched from queues using a fair queuing technique so that, for example, a poorly-behaved controller does not cause others (even at the same priority level) to become abnormal. For details, see **API Priority and Fairness**.

- **APIListChunking**

  APIListChunking moved to GA. This feature allows clients to perform pagination in List requests to avoid performance problems caused by returning too much data at a time.

- lastPhaseTransitionTime of PersistentVolume (PV)

lastPhaseTransitionTime moved to beta. With this feature enabled, Kubernetes adds the **lastPhaseTransitionTime** field to the **status** field of a PV to indicate the time when the PV phase changes last time. Cluster administrators are now able to track the last time a PV transitioned to a different phase, allowing for more efficient and informed resource management. For details, see **PersistentVolume Last Phase Transition Time in Kubernetes**.

- **ReadWriteOncePod**

  ReadWriteOncePod moved to GA. With this feature enabled, you can set the access mode to **ReadWriteOncePod** in a PersistentVolumeClaim (PVC) to ensure that only one pod can modify data in the volume at a time. This can prevent data conflicts or damage. For details, see **ReadWriteOncePod**.

- **CSINodeExpandSecret**

  CSINodeExpandSecret moved to GA. This feature allows secret authentication data to be passed to a CSI driver for use when a node is added.

- CEL-based CustomResourceDefinition (CRD) verification

  The CEL-based CRD verification capability moved to GA. With this feature enabled, you are allowed to use the Common Expression Language (CEL) to define validation rules in CRDs, which are more efficient than webhook. For details, see **CRD verification rules**.

API Changes and Removals

Kubernetes 1.31

In Kubernetes 1.31, if **caBundle** is not empty but the value is invalid or it does not define any CA certificate, the CRD does not provide services. If **caBundle** is set to a valid value, it remains unchanged if updated. Attempting direct updates results in an "invalid field value" error, ensuring uninterrupted CRD services.

Kubernetes 1.30

- kubectl replaces **prune-whitelist** with **prune-allowlist** in the **apply** command.
- SecurityContextDeny, which has been deprecated in Kubernetes 1.27, is replaced by **Pod Security Admission**.

Kubernetes 1.29

- The time zone of a newly created CronJob cannot be configured using **TZ** or **CRON_TZ** in **.spec.schedule**. Use **.spec.timeZone** instead. CronJobs that have been created are not affected by this change.
- The alpha API **ClusterCIDR** is removed.
- The startup parameter **--authentication-config** is added to kube-apiserver to specify the address of the **AuthenticationConfiguration** file. This startup parameter is mutually exclusive with the **--oidc-*** startup parameter.
- The API version **kubescheduler.config.k8s.io/v1beta3** of **KubeSchedulerConfiguration** is removed. Migrate **kube-scheduler** configuration files to **kubescheduler.config.k8s.io/v1**.
- The CEL expressions are added to **v1alpha1 AuthenticationConfiguration**.

- **ServiceCIDR** is added. It allows you to specify a CIDR block for a ClusterIP Service.
- The startup parameters **--conntrack-udp-timeout** and **--conntrack-udp-timeout-stream** are added to **kube-proxy**. They are options for configuring the kernel parameters **nf_conntrack_udp_timeout** and **nf_conntrack_udp_timeout_stream**.
- CEL expressions are supported by **WebhookMatchCondition** of **v1alpha1 AuthenticationConfiguration**.
- The type of **PVC.spec.Resource** is changed from **ResourceRequirements** to **VolumeResourceRequirements**.
- **onPodConditions** in **PodFailurePolicyRule** is marked as optional.
- The API version **flowcontrol.apiserver.k8s.io/v1beta3** of **FlowSchema** and **PriorityLevelConfiguration** has been upgraded to **flowcontrol.apiserver.k8s.io/v1**, and the following changes have been made:
  - **PriorityLevelConfiguration**:
    The **.spec.limited.nominalConcurrencyShares** field defaults to **30** if the field is omitted. To ensure compatibility with 1.28 API servers, specifying an explicit **0** is not allowed in the **v1** version in 1.29. In 1.30, explicit **0** will be allowed in this field in the **v1** API. The **flowcontrol.apiserver.k8s.io/v1beta3** APIs are deprecated and will no longer be served in 1.32.
- The **kube-proxy** command line document is updated. **kube-proxy** does not bind any socket to the IP address specified by **--bind-address**.
- If **CSI-Node-Driver** is not running, **NodeStageVolume** calls will be retried.
- **ValidatingAdmissionPolicy** type checking now supports CRDs. To use this feature, the **ValidatingAdmissionPolicy** feature gate must be enabled.
- The startup parameter **--nf-conntrack-tcp-be-liberal** is added to **kube-proxy**. You can configure it by setting the kernel parameter **nf_conntrack_tcp_be_liberal**.
- The startup parameter **--init-only** is added to **kube-proxy**. Setting the flag makes **kube-proxy** init container run in the privileged mode, perform its initial configuration, and then exit.
- The **fileSystem** field of container is added to the response body of CRI. It specifies the file system usage of a container. Originally, the **fileSystem** field contains only the file system of the container images.
- All built-in cloud providers are disabled by default. If you still need to use them, you can configure the **DisableCloudProviders** and **DisableKubeletCloudCredentialProvider** feature gates to disable or enable cloud providers.

References

For more details about the performance comparison and function evolution between Kubernetes 1.31 and other versions, see the following documents:

- **Kubernetes v1.31 Release Notes**
- **Kubernetes v1.30 Release Notes**
- **Kubernetes v1.29 Release Notes**

4.1.2.2 Kubernetes 1.28 Release Notes

CCE has passed the Certified Kubernetes Conformance Program and is a certified Kubernetes offering. CCE allows you to create clusters of Kubernetes 1.28. This topic describes the changes made in Kubernetes 1.28.

Indexes

Important Notes

- In Kubernetes 1.28, the scheduling framework is improved to reduce useless retries. The overall scheduling performance is enhanced. If a custom scheduler plugin is used in a cluster, you can perform the adaptation upgrade following the instructions in **GitHub**.
- The Ceph FS in-tree volume plugin has been deprecated in Kubernetes 1.28 and will be removed in Kubernetes 1.31. (The community does not plan to support CSI migration.) Use **Ceph CSI driver** instead.
- The Ceph RBD in-tree volume plugin has been deprecated in Kubernetes 1.28 and will be removed in Kubernetes 1.31. (The community does not plan to support CSI migration.) Use RBD **Ceph CSI driver** instead.

New and Enhanced Features

Features in alpha stage are disabled by default, those in beta stage are enabled by default, and those in GA stage are always enabled and they cannot be disabled. The function of turning on or off the features in GA stage will be removed in later Kubernetes versions. CCE policies for new features are the same as those in the community.

- The version skew policy is expanded to three versions.

  Starting with control planes 1.28 and worker nodes 1.25, the Kubernetes skew policy expands the supported control plane and worker node skew to three versions. This enables annual minor version upgrades of nodes while staying on supported minor versions. For details, see **Version Skew Policy**.

- Retroactive Default StorageClass moves to GA.

  The retroactive default StorageClass assignment graduates to GA. This enhancement brings a significant improvement to how default StorageClasses are assigned to PersistentVolumeClaims (PVCs).

  The PV controller has been modified to automatically assign a default StorageClass to any unbound PVC with **storageClassName** not configured. Additionally, the PVC admission validation mechanism within the API server has been adjusted to allow changing values from an unset state to an actual StorageClass name. For details, see **Retroactive default StorageClass assignment**.

- Native sidecar containers are introduced.

  The native sidecar containers are available in alpha. Kubernetes 1.28 adds **restartPolicy** to Init containers. This field is available when the SidecarContainers feature gate is enabled. However, there are still some problems to be solved in the native sidecar containers. Therefore, the Kubernetes community recommends only using this feature gate in **short lived testing clusters** at the alpha phase. For details, see **Introducing native sidecar containers**.

- Mixed version proxy is introduced.

  A new mechanism (mixed version proxy) is released to improve cluster upgrade. It is an alpha feature in Kubernetes 1.28. When a cluster undergoes an upgrade, API servers of different versions in the cluster can serve different sets (groups, versions, or resources) of built-in resources. A resource request made in this scenario may be served by any of the available API servers, potentially resulting in the request ending up at an API server that may not be aware of the requested resource. As a result, the request fails. This feature can solve this problem. (Note that CCE provides hitless upgrade. Therefore, this feature is not used in CCE clusters.) For details, see **A New (alpha) Mechanism For Safer Cluster Upgrades**.

- Non-graceful node shutdown moves to GA.

  The non-graceful node shutdown is now GA in Kubernetes 1.28. When a node was shut down and that shutdown was not detected by the kubelet's Node Shutdown Manager, the StatefulSet pods that run on this node will stay in the terminated state and cannot be moved to a running node. If you have confirmed that the shutdown node is unrecoverable, you can add an **out-of-service** taint to the node. This ensures that the StatefulSet pods and VolumeAttachments on this node can be forcibly deleted and the corresponding pods will be created on a healthy node. For details, see **Non-Graceful Node Shutdown Moves to GA**.

- NodeSwap moves to beta.

  Support for NodeSwap goes to beta in Kubernetes 1.28. NodeSwap is disabled by default and can be enabled using the NodeSwap feature gate. NodeSwap allows you to configure swap memory usage for Kubernetes workloads running on Linux on a per-node basis. Note that although NodeSwap has reached beta, there are still some problems to be solved and security risks to be enhanced. For details, see **Beta Support for Using Swap on Linux**.

- Two job-related features are added.

  Two alpha features are introduced: **delayed creation of replacement pods** and **backoff limit per index**.

  - Delayed creation of replacement pods

    By default, when a pod enters the terminating state (for example, due to the preemption or eviction), Kubernetes immediately creates a replacement pod. Therefore, both pods are running concurrently.

    In Kubernetes 1.28, this feature can be enabled by turning on the JobPodReplacementPolicy feature gate. With this feature gate enabled, you can set the **podReplacementPolicy** field under **spec** of a job to **Failed**. In this way, pods would only be replaced when they reached the failed phase, and not when they are terminating. Additionally, you can check the **.status.termination** field of a job. The value of this field is the number of pods owned by the job that are currently terminating.

  - Backoff limit per index

    By default, pod failures for indexed jobs are recorded and restricted by the global limit of retries, specified by **.spec.backoffLimit**. This means that if there is a consistently failing index in a job, pods specified by the job will be restarted repeatedly until pod failures exhaust the limit. Once the limit is reached, the job is marked failed and pods for other indexes in the job may never be even started.

    In Kubernetes 1.28, this feature can be enabled by turning on the JobBackoffLimitPerIndex feature gate of a cluster. With this feature gate enabled, **.spec.backoffLimitPerIndex** can be specified when an indexed job is created. Only if the failures of pods with all indexes specified in this job exceed the upper limit, pods specified by the job will not be restarted.

- Some CEL related features are improved.

  CEL related capabilities are enhanced.

  - CEL used to validate CRDs moves to beta.

    This feature has been upgraded to beta since Kubernetes 1.25. By embedding CEL expressions into CRDs, developers can solve most of the CR validation use cases without using webhooks. More CEL functions, such as support for default value and CRD conversion, will be developed in later Kubernetes versions.

  - CEL admission control graduates to beta.

    CEL admission control is customizable. With CEL expressions, you can decide whether to accept or reject requests received by kube-apiserver. CEL expressions can also serve as a substitute for admission webhooks. Kubernetes 1.28 has upgraded CEL admission control to beta and introduced new functions, such as:

    - ValidatingAdmissionPolicy can correctly handle the **authorizer** variable.
    - ValidatingAdmissionPolicy can have the **messageExpression** field checked.
    - The ValidatingAdmissionPolicy controller is added to kube-controller-manager to check the type of the CEL expression in ValidatingAdmissionPolicy and save the reason in the **status** field.
    - CEL expressions can contain a combination of one or more variables, which can be defined in ValidatingAdmissionPolicy. These variables can be used to define other variables.
    - CEL library functions can be used to parse resources specified by **resource.Quantity** in Kubernetes.

- Other features
  - The ServiceNodePortStaticSubrange feature gate moves to beta. With this feature enabled, static port range can be reserved to avoid conflicts with dynamically allocated ports. For details, see **Avoiding Collisions Assigning Ports to NodePort Services**.
  - The alpha feature ConsistentListFromCache is added to allow the API server to serve consistent lists from cache. Get and list requests can read data from the cache instead of etcd.

  - In Kubernetes 1.28, kubelet can configure the drop-in directory (alpha). This feature allows you to add support for the **--config-dir** flag to kubelet so that you can specify an insert directory that overwrites the kubelet configuration in **/etc/kubernetes/kubelet.conf**.
  - ExpandedDNSConfig moves to GA and is enabled by default. With this feature enabled, DNS configurations can be expanded.
  - The alpha feature CRDValidationRatcheting is added. This feature allows CRs with failing validations to pass if a Patch or Update request does not alter any of the invalid fields.
  - **--concurrent-cron-job-syncs** is added to kube-controller-manager to configure the number of workers for the cron job controller.

API Changes and Removals

- **NetworkPolicyStatus** is removed. There is no status attribute in a network policy.
- **annotationbatch.kubernetes.io/cronJob-scheduled-timestamp** is added to job objects to indicate the creation time of a job.
- The **podReplacementPolicy** and **terminating** fields are added to job APIs. With these fields specified, once a previously created pod is terminated in a job, the job immediately starts a new pod to replace the pod. The new fields allow you to specify whether to replace the pod immediately after the previous pod is terminated (original behavior) or replace the pod after the existing pod is completely terminated (new behavior). This is an alpha feature, and you can enable it by turning on the **JobPodReplacementPolicy** feature gate in your cluster.
- The **BackoffLimitPerIndex** field is available in a job. Pods specified by a job share a backoff mechanism. When backoff times of the job reach the limit, this job is marked as failed and resources, including indexes that are not running, are cleared up. This field allows you to configure backoff limit for a single index. For details, see **Backoff limit per index**.
- The **ServedVersions** field is added to the **StorageVersion** API. This change is introduced by mixed version proxy. The new field is used to indicate a version that can be provided by the API server.
- **SelfSubjectReview** is added to **authentication.k8s.io/v1**, and **kubectl auth whoami** goes to GA.
- **LastPhaseTransitionTime** is added to **PersistentVolume**. The new field is used to store the last time when a volume changes to a different phase.
- **resizeStatus** in **PVC.Status** is replaced by **AllocatedResourceStatus**. The new field indicates the statuses of the storage resize operation. The default value is an empty string.
- If **hostNetwork** is set to **true** and ports are specified for a pod, the **hostport** field will be automatically configured.
- StatefulSet pods have the pod index set as a pod label **statefulset.kubernetes.io/pod-index**.
- **PodHasNetwork** in the **Condition** field of pods has been renamed to **PodReadyToStartContainers**. The new field specifies that containers are ready to start after the network, volumes, and sandbox pod have been created.

- A new configuration option **delayCacheUntilActive** is added to **KubeSchedulerConfiguration**. If **delayCacheUntilActive** is set to **true**, kube-scheduler on the leader will not cache scheduling information. This reduces the memory pressure of other master nodes, but slows down the failover speed after the leader failed.
- The **namespaceParamRef** field is added to **admissionregistration.k8s.io/v1alpha1.ValidatingAdmissionPolicy**.
- The **reason** and **fieldPath** fields are added to CRD validation rules to allow you to specify reason and field path after verification failed.
- The CEL expression of ValidatingAdmissionPolicy supports namespace access via namespaceObject.
- API groups ValidatingAdmissionPolicy and ValidatingAdmissionPolicyBinding are promoted to betav1.
- A ValidatingAdmissionPolicy now has its **messageExpression** field checked against resolved types.

Feature Gate and Command Line Parameter Changes and Removals

- **–short** is removed from kubelet. Therefore, the default output of **kubectl version** is the same as that of **kubectl version –short**.
- **--volume-host-cidr-denylist** and **--volume-host-allow-local-loopback** are removed from kube-controller-manager. **--volume-host-cidr-denylist** is a comma-separated list of CIDR ranges. Volume plugins at these IP addresses are not allowed. If **--volume-host-allow-local-loopback** is set to **false**, the local loopback IP address and the CIDR ranges specified in **--volume-host-cidr-denylist** are disabled.
- **--azure-container-registry-config** is deprecated in kubelet and will be deleted in later Kubernetes versions. Use **--image-credential-provider-config** and **--image-credential-provider-bin-dir** instead.
- **--lock-object-namespace** and **--lock-object-name** are removed from kube-scheduler. Use **--leader-elect-resource-namespace** and **--leader-elect-resource-name** or **ComponentConfig** instead. (**--lock-object-namespace** is used to define the namespace of a lock object, and **--lock-object-name** is used to define the name of a lock object.)
- KMS v1 is deprecated and will only receive security updates. Use KMS v2 instead. In later Kubernetes versions, use **--feature-gates=KMSv1=true** to configure a KMS v1 provider.
- The DelegateFSGroupToCSIDriver, DevicePlugins, KubeletCredentialProviders, MixedProtocolLBService, ServiceInternalTrafficPolicy, ServiceIPStaticSubrange, and EndpointSliceTerminatingCondition feature gates are removed.

References

For more details about the performance comparison and function evolution between Kubernetes 1.28 and other versions, see **Kubernetes v1.28 Release Notes**.

4.1.2.3 Kubernetes 1.27 Release Notes

CCE has passed the Certified Kubernetes Conformance Program and is a certified Kubernetes offering. CCE allows you to create clusters of Kubernetes 1.27. This topic describes the changes made in Kubernetes 1.27.

Indexes

New Features

- SeccompDefault is stable.

  To use SeccompDefault, add the **--seccomp-default** **command line flag** using kubelet on each node. If this feature is enabled, the **RuntimeDefault** profile will be used for all workloads by default, instead of the **Unconfined** (seccomp disabled) profile.

- Jobs' scheduling directives are configurable.

  This feature was introduced in Kubernetes 1.22 and is stable in Kubernetes 1.27. In most cases, you use a job to influence where the pods will run, like all in the same AZ. This feature allows scheduling directives to be modified before a job starts. You can use the **suspend** field to suspend a job. In the suspension phase, the scheduling directives (such as the node selector, node affinity, anti-affinity, and tolerations) in the job's pod template can be modified. For details, see **Mutable Scheduling Directives**.

- Downward API hugepages are stable.

  In Kubernetes 1.20, **requests.hugepages-***<pagesize>* and **limits.hugepages-***<pagesize>* were introduced to the **downward API**. Requests and limits can be configured for hugepages like other resources.

- Pod scheduling readiness moves to beta.

  After a pod is created, the Kubernetes scheduler selects an appropriate node to run the pod in the pending state. In practice, some pods may stay in the pending state for a long period due to insufficient resources. These pods may affect the running of other components like Cluster Autoscaler in the cluster. By specifying or deleting **.spec. schedulingGates** for a pod, you can control when the pod is ready for scheduling. For details, see **Pod Scheduling Readiness**.

- Accessing node logs using Kubernetes APIs is supported.

  This function is in the alpha phase. The cluster administrator can directly query node logs to help debug malfunctioning services running on the node. To use this function, ensure that the NodeLogQuery **feature gate** is enabled for that node and the kubelet configuration options **enableSystemLogHandler** and **enableSystemLogQuery** are set to **true**.

- ReadWriteOncePod access mode moves to beta.

  Kubernetes 1.22 introduced a ReadWriteOncePod access mode for PVs and PVCs. This feature has evolved into the beta phase. A volume can be mounted to a single pod in read/write mode. Use this access mode if you want to

ensure that only one pod in the cluster can read that PVC or write to it. For details, see **Access Modes**.

- The **matchLabelKeys** field in the pod topology spread constraint moves to beta.

  matchLabelKeys is a list of pod label keys. It is used to select a group of pods over which spreading will be calculated. With **matchLabelKeys**, you do not need to update **pod.spec** between different revisions. The controller or operator just needs to set different values to the same label key for different revisions. The scheduler will automatically determine the values based on **matchLabelKeys**. For details, see **Pod Topology Distribution Constraints**.

- The function of efficiently labeling SELinux volumes moves to beta.

  By default, the container runtime recursively assigns the SELinux label to all files on all pod volumes. To speed up this process, Kubernetes uses the mount option **-o context=***<label>* to immediately change the SELinux label of the volume. For details, see **Efficient SELinux volume relabeling**.

- VolumeManager reconstruction goes to beta.

  After the VolumeManager is reconstructed, if the NewVolumeManagerReconstruction **feature gate** is enabled, mounted volumes will be obtained in a more effective way during kubelet startup.

- Server side field validation and OpenAPI V3 are stable.

  OpenAPI V3 was added in Kubernetes 1.23. In Kubernetes 1.24, it moved to beta. In Kubernetes 1.27, it is stable.

- StatefulSet start ordinal moves to beta.

  Kubernetes 1.26 introduced a new, alpha-level feature for StatefulSets to control the ordinal numbering of pod replicas. Since Kubernetes 1.27, this feature moves to beta. The ordinals can start from arbitrary non-negative numbers. For details, see **Kubernetes 1.27: StatefulSet Start Ordinal Simplifies Migration**.

- **ContainerResource** metric in HorizontalPodAutoscaler moves to beta.

  Kubernetes 1.20 introduced the **ContainerResource** metric in HorizontalPodAutoscaler (HPA). In Kubernetes 1.27, this feature moves to beta, and the HPAContainerMetrics feature gate is enabled by default.

- StatefulSet PVC auto deletion moves to beta.

  Kubernetes 1.27 provides a new policy to control the lifecycle of PVCs of StatefulSets. This policy allows users to specify if the PVCs generated from the StatefulSet spec template should be automatically deleted or retained when the StatefulSet is deleted or replicas in the StatefulSet are scaled down. For details, see **PersistentVolumeClaim retention**.

- Volume group snapshots are introduced.

  Volume group snapshots are introduced as an alpha feature in Kubernetes 1.27. This feature allows users to create snapshots for multiple volumes to ensure data consistency when a fault occurs. It uses a label selector to group multiple PVCs for snapshot. This feature only supports CSI volume drivers. For details, see **Kubernetes 1.27: Introducing an API for Volume Group Snapshots**.

- **kubectl apply** pruning is more secure and efficient.

  In Kubernetes 1.5, the **--prune** flag was introduced in **kubectl apply** to delete resources that are no longer needed. This allowed **kubectl apply** to

automatically clear resources removed from the current configuration. However, the existing implementation of **--prune** has design defects that degrade its performance and lead to unexpected behaviors. In Kubernetes 1.27, **kubectl apply** provides ApplySet-based pruning, which is in the alpha phase. For details, see **Declarative Management of Kubernetes Objects Using Configuration Files**.

- Conflicts during port allocation to NodePort Service can be avoided.

  In Kubernetes 1.27, you can enable a new **feature gate** ServiceNodePortStaticSubrange to use different port allocation policies for NodePort Services. This mitigates the risk of port conflicts. This feature is in the alpha phase.

- Resizing resources assigned to pods without restarting the containers is supported.

  Kubernetes 1.27 allows users to resize CPU and memory resources assigned to pods without restarting the container. This feature is in the alpha phase. For details, see **Kubernetes 1.27: In-place Resource Resize for Kubernetes Pods (alpha)**.

- Pod startup is accelerated.

  A series of parameter adjustments like parallel image pulls and increased default API query limit for kubelet per second are made in Kubernetes 1.27 to accelerate pod startup. For details, see **Kubernetes 1.27: updates on speeding up Pod startup**.

- KMS V2 moves to beta.

  The key management KMS V2 API goes to beta. This has greatly improved the performance of the KMS encryption provider. For details, see **Using a KMS provider for data encryption**.

Deprecations and Removals

- In Kubernetes 1.27, the feature gates that are used for volume extension and in the GA status, including ExpandCSIVolumes, ExpandInUsePersistentVolumes, and ExpandPersistentVolumes are removed and can no longer be referenced in the **--feature-gates** flag.
- The **--master-service-namespace** parameter is removed. This parameter specifies where to create a Service named **kubernetes** to represent the API server. This parameter was deprecated in Kubernetes 1.26 and is removed from Kubernetes 1.27.
- The ControllerManagerLeaderMigration feature gate is removed. **Leader Migration** provides a mechanism for HA clusters to safely migrate "cloud specific" controllers using a resource lock shared between kube-controller-manager and cloud-controller-manager when upgrading the replicated control plane. This feature has been enabled unconditionally since its release in Kubernetes 1.24. In Kubernetes 1.27, this feature is removed.
- The **--enable-taint-manager** parameter is removed. The feature that it supports, taint-based eviction, is enabled by default and will continue to be implicitly enabled when the flag is removed.
- The **--pod-eviction-timeout** parameter is removed from kube-controller-manager.

- The CSIMigration feature gate is removed. The **CSI migration** program allows smooth migration from the in-tree volume plug-ins to the out-of-tree CSI drivers. This feature was officially released in Kubernetes 1.16.
- The CSIInlineVolume feature gate is removed. The feature (**CSI Ephemeral Volume**) allows CSI volumes to be specified directly in the pod specification for ephemeral use cases. They can be used to inject arbitrary states, such as configuration, secrets, identity, variables, or similar information, directly inside the pod using a mounted volume. This feature graduated to GA in Kubernetes 1.25 and is removed in Kubernetes 1.27.
- The EphemeralContainers feature gate is removed. For Kubernetes 1.27, API support for ephemeral containers is unconditionally enabled.
- The LocalStorageCapacityIsolation feature gate is removed. This feature gate (**Local Ephemeral Storage Capacity Isolation**) moved to GA in Kubernetes 1.25. The feature provides support for capacity isolation of local ephemeral storage between pods, such as emptyDir volumes, so that a pod can be limited in its consumption of shared resources. kubelet will evict a pod if its consumption of local ephemeral storage exceeds the configured limit.
- The NetworkPolicyEndPort feature gate is removed. In Kubernetes 1.25, **endPort** in NetworkPolicy moved to GA. NetworkPolicy providers that support the **endPort** field can be used to specify a range of ports to apply NetworkPolicy.
- The StatefulSetMinReadySeconds feature gate is removed. For a pod that is part of a StatefulSet, Kubernetes marks the pod as read-only when the pod is available (and passes the check) at least within the period specified in **minReadySeconds**. This feature was officially released in Kubernetes 1.25. It is locked to **true** and removed from Kubernetes 1.27.
- The IdentifyPodOS feature gate is removed. If this feature is enabled, you can specify an OS for a pod. It has been stable since Kubernetes 1.25. This feature is removed from Kubernetes 1.27.
- The DaemonSetUpdateSurge feature gate is removed. In Kubernetes 1.25, this feature was stable. It was implemented to minimize DaemonSet downtime during deployment, but it is removed from Kubernetes 1.27.
- The **--container-runtime** parameter is removed. kubelet accepts a deprecated parameter **--container-runtime**, and the only valid value will be **remote** after the dockershim code is removed. This parameter was deprecated in 1.24 and later versions and is removed from Kubernetes 1.27.

References

For more details about the performance comparison and function evolution between Kubernetes 1.27 and other versions, see **Kubernetes v1.27 Release Notes**.

4.1.3 CCE Autopilot Cluster Patch Release History

Indexes

- **v1.31 (OBT)**
- **v1.28**
- **v1.27**

v1.31 (OBT)

Table 4-1 Release notes for the v1.31 patch

CCE Autopilot Cluster Patch Version	Kubernetes Version	Feature Update	Enhancement	Vulnerability Fixing
v1.31.1-r0	v1.31.1	Supported cluster v1.31. For more information, see **Kubernetes 1.31 Release Notes**.	-	Fixed some security issues.

v1.28

Table 4-2 Release notes for the v1.28 patch

CCE Autopilot Cluster Patch Version	Kubernetes Version	Feature Update	Enhancement	Vulnerability Fixing
v1.28.7-r0	v1.28.3	• Allowed namespaces or workloads to have subnets bound. • Allowed creation of subdirectories for PVs that are dynamically provisioned from SFS Turbo file systems.	-	Fixed some security issues.
v1.28.6-r0	v1.28.3	• Supported EVS volumes. • Allowed namespaces or workloads to have security groups and subnets bound. • Supported custom disk storage capacity.	• Enhanced cluster monitoring for better O&M. • Reduced pod startup time. • Optimized the performance of large-scale clusters.	Fixed some security issues.

CCE Autopilot Cluster Patch Version	Kubernetes Version	Feature Update	Enhancement	Vulnerability Fixing
v1.28.5-r0	v1.28.3	• Supported APM probes during workload creation. • Supported access to kube-apiserver using a private IP address.	-	Fixed some security issues.
v1.28.4-r0	v1.28.3	• Custom metrics, such as network and disk metrics, can be used create HPA policies. • kube-apiserver can be accessed from a public network.	When YAML is used to create an application, the parameters that are not supported by CCE Autopilot and do not affect functionality were automatically ignored.	Fixed some security issues.
v1.28.3-r0	v1.28.3	• Hosted the Everest storage add-on at the backend. • Supported OBS volumes.	-	Fixed some security issues.
v1.28.2-r0	v1.28.3	• Supported CronHPA policies. • Supported the security context configuration for pods.	-	Fixed some security issues.
v1.28.1-r10	v1.28.3	Supported cluster v1.28. For more information, see **Kubernetes 1.28 Release Notes**.	-	-

v1.27

Table 4-3 Release notes for the v1.27 patch

CCE Autopilot Cluster Patch Version	Kubernetes Version	Feature Update	Enhancement	Vulnerability Fixing
v1.27.9-r0	v1.27.4	• Allowed namespaces or workloads to have subnets bound. • Allowed creation of subdirectories for PVs that are dynamically provisioned from SFS Turbo file systems.	-	Fixed some security issues.
v1.27.8-r0	v1.27.4	• Supported EVS volumes. • Allowed namespaces or workloads to have security groups and subnets bound. • Supported custom disk storage capacity.	• Enhanced cluster monitoring for better O&M. • Reduced pod startup time. • Optimized the performance of large-scale clusters.	Fixed some security issues.
v1.27.7-r0	v1.27.4	• Supported APM probes during workload creation. • Supported access to kube-apiserver using a private IP address.	-	Fixed some security issues.
v1.27.6-r0	v1.27.4	• Custom metrics, such as network and disk metrics, can be used create HPA policies. • kube-apiserver can be accessed from a public network.	When YAML is used to create an application, the parameters that are not supported by CCE Autopilot and do not affect functionality were automatically ignored.	Fixed some security issues.
v1.27.5-r0	v1.27.4	• Hosted the Everest storage add-on at the backend. • Supported OBS volumes.	-	Fixed some security issues.

CCE Autopilot Cluster Patch Version	Kubernetes Version	Feature Update	Enhancement	Vulnerability Fixing
v1.27.4-r0	v1.27.4	• Supported CronHPA policies. • Supported the security context configuration for pods.	-	Fixed some security issues.
v1.27.3-r30	v1.27.4	-	Supported one-click configuration of monitoring alarms.	Fixed some security issues.
v1.27.3-r20	v1.27.4	• Supported the Nginx Ingress Controller add-on. • Supported the Cloud Native Cluster Monitoring and Cloud Native Logging add-ons to monitor application metrics and collect application logs. • Launched the application template market. • Supported CustomResourceDefinitions (CRDs). • Interconnected with CloudShell.	Optimized the function of creating a NAT gateway by default during cluster creation so that applications can access the public network.	Fixed some security issues.
v1.27.3-r10	v1.27.4	Supported cluster v1.27. For more information, see **Kubernetes 1.27 Release Notes**.	-	-

4.2 Add-on Versions

4.2.1 CoreDNS Release History

Table 4-4 Release history

Add-on Version	Supported Cluster Version	New Feature	Community Version
1.28.10	• v1.31 • v1.28 • v1.27	Supported CCE Autopilot clusters v1.31.	1.10.1
1.28.9	• v1.28 • v1.27	Fixed some issues.	1.10.1
1.28.8	• v1.28 • v1.27	Performed a regular upgrade of add-on dependencies.	1.10.1
1.28.7	• v1.28 • v1.27	Performed a regular upgrade of add-on dependencies.	1.10.1
1.28.6	• v1.28 • v1.27	Fixed some issues.	1.10.1

4.2.2 NGINX Ingress Controller Release History

Table 4-5 Release history for NGINX Ingress Controller 2.4.x

Add-on Version	Supported Cluster Version	New Feature	Community Version
2.4.14	• v1.31 • v1.28 • v1.27	• Supported CCE Autopilot clusters v1.31. • Fixed the CVE-2024-7646 vulnerability.	1.11.2
2.4.13	• v1.28 • v1.27	Supported the community version v1.9.6.	1.9.6
2.4.12	• v1.28 • v1.27	Fixed some issues.	1.9.3
2.4.11	• v1.28 • v1.27	Fixed some issues.	1.9.3
2.4.10	• v1.28 • v1.27	Fixed some issues.	1.9.3

4.2.3 Kubernetes Metrics Server Release History

Table 4-6 Release history

Add-on Version	Supported Cluster Version	New Feature	Community Version
1.3.42	• v1.31 • v1.28 • v1.27	Supported CCE Autopilot clusters v1.31.	**0.6.2**
1.3.41	• v1.28 • v1.27	Fixed some issues.	**0.6.2**
1.3.40	• v1.28 • v1.27	Fixed some issues.	**0.6.2**
1.3.39	• v1.28 • v1.27	Fixed some issues.	**0.6.2**
1.3.38	• v1.28 • v1.27	• Supported anti-affinity scheduling of add-on pods on nodes in different AZs. • Changed the default taint tolerance duration to 60s. • Synchronized time zones used by the add-on and the node.	**0.6.2**

4.2.4 CCE Advanced HPA Release History

Table 4-7 Release history

Add-on Version	Supported Cluster Version	New Feature
1.3.48	• v1.31 • v1.28 • v1.27	Supported CCE Autopilot clusters v1.31.
1.3.47	• v1.28 • v1.27	Fixed some issues.
1.3.46	• v1.28 • v1.27	Performed a regular upgrade of add-on dependencies.
1.3.45	• v1.28 • v1.27	Fixed some issues.

Add-on Version	Supported Cluster Version	New Feature
1.3.44	• v1.28 • v1.27	Supported CronHPA policies.

4.2.5 Cloud Native Cluster Monitoring Release History

Table 4-8 Release history

Add-on Version	Supported Cluster Version	New Feature	Community Version
3.12.0	• v1.31 • v1.28 • v1.27	Supported CCE Autopilot clusters v1.31.	2.53.2
3.9.6	• v1.28 • v1.27	Upgraded the Prometheus version and removed the node-exporter component.	2.53.2
3.9.5	• v1.28 • v1.27	Supported custom metrics.	2.37.8
3.9.3	• v1.28 • v1.27	Fixed some issues.	2.37.8

4.2.6 Cloud Native Log Collection Release History

Table 4-9 Release history

Add-on Version	Supported Cluster Version	New Feature
1.7.0	• v1.31 • v1.28 • v1.27	Supported CCE Autopilot clusters v1.31.
1.4.5	• v1.28 • v1.27	Optimized the resource configuration.
1.4.4	• v1.28 • v1.27	Fixed some issues.

Add-on Version	Supported Cluster Version	New Feature
1.4.3	• v1.28 • v1.27	Fixed some issues.
1.2.25	• v1.28 • v1.27	Fixed some issues.

