

Cloud Container Engine

API Reference

Issue 01
Date 2024-05-31



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2024. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Cloud Computing Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI 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.

Contents

1 Before You Start.....	1
1.1 Overview.....	1
1.2 API Calling.....	2
1.3 Endpoints.....	2
1.4 Constraints.....	2
1.5 Concepts.....	3
2 API Overview.....	5
3 Calling APIs.....	23
3.1 Making an API Request.....	23
3.2 Authentication.....	27
3.3 Response.....	29
4 APIs.....	31
4.1 API URL.....	31
4.2 Cluster Management.....	32
4.2.1 Creating a Cluster.....	32
4.2.2 Reading a Specified Cluster.....	96
4.2.3 Listing Clusters in a Specified Project.....	117
4.2.4 Updating a Specified Cluster.....	140
4.2.5 Deleting a Cluster.....	176
4.2.6 Hibernating a Cluster.....	203
4.2.7 Waking Up a Cluster.....	207
4.2.8 Obtaining a Cluster Certificate.....	211
4.2.9 Modifying Cluster Specifications.....	218
4.2.10 Querying a Job.....	228
4.2.11 Binding/Unbinding Public API Server Address.....	234
4.2.12 Obtaining Cluster Access Address.....	242
4.2.13 Obtaining a Cluster's Logging Configurations.....	247
4.2.14 Configuring Cluster Logs.....	252
4.2.15 Obtaining the Partition List.....	258
4.2.16 Creating a Partition.....	263
4.2.17 Obtaining Partition Details.....	270
4.2.18 Updating a Partition.....	275

4.3 Node Management.....	282
4.3.1 Creating a Node.....	282
4.3.2 Reading a Specified Node.....	359
4.3.3 Listing All Nodes in a Cluster.....	387
4.3.4 Updating a Specified Node.....	415
4.3.5 Deleting a Node.....	443
4.3.6 Accepting a Node.....	471
4.3.7 Resetting a Node.....	495
4.3.8 Removing a Node.....	523
4.3.9 Migrating a Node.....	532
4.4 Node Pool Management.....	550
4.4.1 Creating a Node Pool.....	550
4.4.2 Reading a Specified Node Pool.....	613
4.4.3 Listing All Node Pools in a Specified Cluster.....	643
4.4.4 Updating a Specified Node Pool.....	672
4.4.5 Deleting a Node Pool.....	714
4.5 Storage Management.....	744
4.5.1 Creating a PVC (to be discarded).....	744
4.5.2 Deleting a PVC (to be discarded).....	754
4.6 Add-on Management.....	761
4.6.1 Installing an Add-on Instance.....	761
4.6.2 Listing Add-on Templates.....	775
4.6.3 Updating an Add-on Instance.....	783
4.6.4 Rolling Back an Add-on Instance.....	798
4.6.5 Deleting an Add-on Instance.....	809
4.6.6 Querying an Add-on Instance.....	813
4.6.7 Listing Add-on Instances.....	821
4.7 Cluster Upgrade.....	829
4.7.1 Upgrading a Cluster.....	829
4.7.2 Obtaining Cluster Upgrade Task Details.....	838
4.7.3 Retrying a Cluster Upgrade Task.....	843
4.7.4 Suspending a Cluster Upgrade Task (Deprecated).....	847
4.7.5 Continuing to Execute a Cluster Upgrade Task (Deprecated).....	850
4.7.6 Obtaining a List of Cluster Upgrade Task Details.....	853
4.7.7 Pre-upgrade Check.....	859
4.7.8 Obtaining Details About a Pre-upgrade Check Task of a Cluster.....	868
4.7.9 Obtaining a List of Pre-upgrade Check Tasks of a Cluster.....	876
4.7.10 Post-upgrade Check.....	885
4.7.11 Cluster Backup.....	888
4.7.12 Obtaining a List of Cluster Backup Task Details.....	890
4.7.13 Obtaining the Cluster Upgrade Information.....	895
4.7.14 Obtaining a Cluster Upgrade Path.....	899

4.7.15 Obtaining the Configuration of Cluster Upgrade Feature Gates.....	901
4.7.16 Enabling the Cluster Upgrade Process Booting Task.....	904
4.7.17 Obtaining a List of Upgrade Workflows.....	912
4.7.18 Obtaining Details About a Specified Cluster Upgrade Task.....	919
4.7.19 Updating the Status of a Specified Cluster Upgrade Booting Task.....	926
4.8 Quota Management.....	935
4.8.1 Querying Resource Quotas.....	935
4.9 API Versions.....	939
4.9.1 Obtaining API Versions.....	939
4.10 Tag Management.....	943
4.10.1 Adding Resource Tags to a Specified Cluster in Batches.....	943
4.10.2 Deleting Resource Tags of a Specified Cluster in Batches.....	949
4.11 Configuration Management.....	954
4.11.1 Obtaining the Parameters That Can Be Configured for a Node Pool.....	954
4.11.2 Obtaining the List of Parameters That Can Be Configured for a Cluster.....	959
4.11.3 Obtaining the Parameters That Can Be Configured for a Node Pool.....	963
4.11.4 Changing the Values of Configuration Parameters of a Node Pool.....	969
4.12 Chart Management.....	978
4.12.1 Uploading a Chart.....	979
4.12.2 Obtaining a Chart List.....	983
4.12.3 Obtaining a Release List.....	988
4.12.4 Updating a Chart.....	992
4.12.5 Creating a Release.....	995
4.12.6 Deleting a Chart.....	1002
4.12.7 Updating a Release.....	1005
4.12.8 Obtaining a Chart.....	1012
4.12.9 Deleting a Release.....	1016
4.12.10 Downloading a Chart.....	1020
4.12.11 Obtaining a Release.....	1024
4.12.12 Obtaining Chart Values.....	1028
4.12.13 Obtaining Historical Records of a Release.....	1032
4.12.14 Obtaining the Quota of a User Chart.....	1037
4.13 Add-on Instance Parameter Description.....	1041
4.13.1 virtual-kubelet.....	1041
5 Kubernetes APIs.....	1045
6 Out-of-Date APIs.....	1050
6.1 Obtaining Cluster Certificates.....	1050
6.2 Creating a PersistentVolume.....	1054
6.3 Deleting a PersistentVolume.....	1066
7 Permissions and Supported Actions.....	1074
8 Appendix.....	1081

8.1 Status Code.....	1081
8.2 Error Codes.....	1085
8.3 Obtaining a Project ID.....	1090
8.4 Obtaining an Account ID.....	1091
8.5 Specifying Add-ons to Be Installed During Cluster Creation.....	1092
8.6 How to Obtain Parameters in the API URI.....	1101
8.7 Creating a VPC and Subnet.....	1103
8.8 Creating a Key Pair.....	1104
8.9 Node Flavor Description.....	1105
8.10 Adding a Salt in the password Field When Creating a Node.....	1108
8.11 Maximum Number of Pods That Can Be Created on a Node.....	1110
8.12 Node OS.....	1112
8.13 Data Disk Space Allocation.....	1122
8.14 Attaching Disks to a Node.....	1128

1 Before You Start

1.1 Overview

Cloud Container Engine (CCE) is a container service that allows you to run containers efficiently in the cloud. CCE provides highly scalable, high-performance, enterprise-class Kubernetes clusters and supports Docker containers. With CCE, you can easily deploy, manage, and scale containerized applications in the cloud.

This document describes how to use APIs for performing operations on CCE, such as creating or deleting CCE resources, modifying resource specifications, or adding NICs. For details about all supported operations, see [API Overview](#).

If you plan to access CCE resources through an API, ensure that you are familiar with CCE concepts. For details, see [Service Overview](#).

CCE supports both Kubernetes-native APIs and proprietary APIs. With these APIs, you can use all functions of CCE.

- CCE has opened APIs through API gateway to support operations on cloud service infrastructures (for example, creating a node). Operations on cluster resources (such as [creating a workload](#)) are also supported.
- Kubernetes-native APIs: You can perform operations on cluster resources (such as [creating a workload](#)) using the Kubernetes-native API server. However, operations on cloud service infrastructures (such as creating a node) are not supported.

For details about Kubernetes-native API versions, see <https://kubernetes.io/docs/concepts/overview/kubernetes-api/>.

NOTE

- The Kubernetes-native APIs called in the current version do not support HTTP persistent connections.
- The Kubernetes-native APIs in the current version include Beta APIs, whose version names include **beta**, for example, **v1beta1**. This type of APIs varies according to Kubernetes-native APIs. Therefore, you are advised to use this type of APIs in unimportant scenarios, for example, short-term test clusters.

1.2 API Calling

CCE supports Representational State Transfer (REST) APIs, allowing you to call APIs using HTTPS. For details about API calling, see [3 Calling APIs](#).

1.3 Endpoints

An endpoint is the **request address** for calling an API. Endpoints vary depending on services and regions. An endpoint can be obtained from [Regions and Endpoints](#).

Select an endpoint based on your service requirements.

- The URL format for cluster, node, node pool, add-on, and quota management is **https://Endpoint/uri**. *uri* indicates the resource path, that is, the API access path.
- The URL format for Kubernetes APIs, storage management, and add-on management is **https://{clusterid}.Endpoint/uri**. In the URL, *{clusterid}* indicates the cluster ID, and *uri* indicates the resource path, that is, the path for API access.

NOTE

- The format of the URL called by the add-on management APIs is **https://{clusterid}.Endpoint/uri**. However, *{clusterid}* is used only for the domain name and is not verified or used by the APIs. Set *{clusterid}* in the query or body. For details about *{clusterid}*, see the add-on management sections.
- *{clusterid}* is required for Kubernetes APIs and storage management, which indicates the cluster that needs to be accessed by calling the API.

Table 1-1 URL parameters

Parameter	Description
{clusterid}	Cluster ID. After a cluster is created, call the API for obtaining a cluster in a specified project to obtain the cluster ID.
Endpoint	Entry (URL) for a web service. Endpoints vary depending on services and regions.
uri	Access path of an API for performing an operation. Obtain the path from the URI of an API. For example, the resource-path of the API used to obtain a user token is v3/auth/tokens .

1.4 Constraints

- CCE imposes a quota on the number and capacity of resources that a user can access. By default, you can create a maximum of five clusters in each region

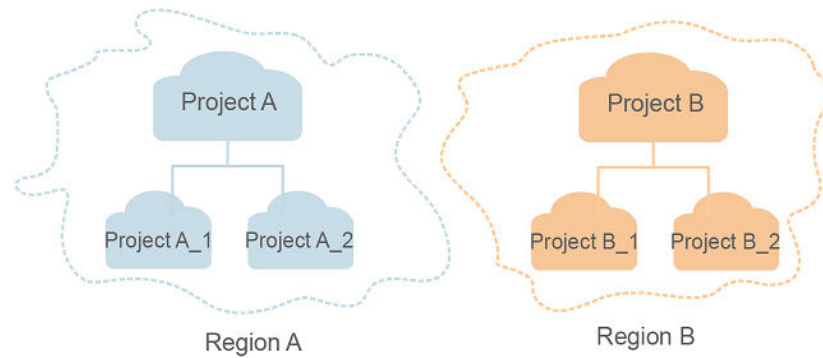
and a cluster can have a maximum of 50 nodes. To create more clusters or add more nodes, [submit a service ticket](#) to increase the quota. For more details about quotas, see [Quotas](#).

- For more constraints, see API description.

1.5 Concepts

- **Account**
An account is created upon successful registration. The account has full access permissions for all of its cloud services and resources. It can be used to reset user passwords and grant user permissions. The account is a payment entity, which should not be used directly to perform routine management. For security purposes, create Identity and Access Management (IAM) users and grant them permissions for routine management.
- **User**
An IAM user is created using an account to use cloud services. Each IAM user has their own identity credentials (password and access keys).
The account name, username, and password will be required for API authentication.
- **Region**
Regions are divided based on geographical location and network latency. Public services, such as Elastic Cloud Server (ECS), Elastic Volume Service (EVS), Object Storage Service (OBS), Virtual Private Cloud (VPC), Elastic IP (EIP), and Image Management Service (IMS), are shared within the same region. Regions are classified into universal regions and dedicated regions. A universal region provides universal cloud services for common tenants. A dedicated region provides specific services for specific tenants.
For details, see [Region and AZ](#).
- **AZ**
An AZ comprises of one or more physical data centers equipped with independent ventilation, fire, water, and electricity facilities. Computing, network, storage, and other resources in an AZ are logically divided into multiple clusters. AZs within a region are interconnected using high-speed optical fibers to allow you to build cross-AZ high-availability systems.
- **Project**
A project corresponds to a region. Default projects are defined to group and physically isolate resources (including compute, storage, and network resources) across regions. Users can be granted permissions in a default project to access all resources under their accounts in the region associated with the project. If you need more refined access control, create subprojects under a default project and create resources in subprojects. Then you can assign users the permissions required to access only the resources in the specific subprojects.

Figure 1-1 Project isolation model



- **Enterprise project**
Enterprise projects group and manage resources across regions. Resources in different enterprise projects are logically isolated. An enterprise project can contain resources of multiple regions, and resources can be added to or removed from enterprise projects.
For details about enterprise projects and about how to obtain enterprise project IDs, see [Enterprise Management User Guide](#).

2 API Overview

APIs provided by CCE are classified into two types: proprietary APIs and Kubernetes-native APIs. By using these two types of APIs, you can use all functions provided by CCE, including creating clusters and nodes, using the Kubernetes-native APIs to create workloads, and using the proprietary CCE APIs to monitor application data.

Type	Subtype	Description
Proprietary CCE APIs	Cluster-related APIs	Manage clusters, including creating and deleting clusters. You can use APIs in this category to create clusters and obtain information about created clusters.
	Node-related APIs	Manage nodes, including creating and deleting nodes. You can use these APIs in this category to add nodes to clusters and obtain information about created nodes.
	Node Pool-related APIs	Manage node pools, including creating and deleting node pools. You can use APIs in this category to create node pools and obtain information about created node pools.
	Add-on-related APIs	Manage add-ons, including querying AddonTemplates and creating, updating, deleting, and obtaining AddonInstances.
	Quota-related API	Query quotas of CCE resources.

Type	Subtype	Description
Kubernetes-native APIs	None	For details about how to call Kubernetes-native APIs, see Kubernetes APIs . NOTE The Kubernetes-native APIs called in the current version do not support HTTP persistent connections.

Cluster-related APIs

Table 2-1 Cluster-related APIs

API	Description
Creating a Cluster	Create an empty cluster, which has only master nodes but do not have worker nodes.
Reading a Specified Cluster	Obtain details about a specified cluster.
Listing Clusters in a Specified Project	Obtain details about all clusters in a specified project.
Updating a Specified Cluster	Update information about a specified cluster.
Deleting a Cluster	Delete a specified cluster.
Hibernating a Cluster	Hibernate a specified cluster.
Waking Up a Cluster	Wake up a hibernated cluster.
Obtaining Cluster Certificates	Obtain certificates of a specified cluster.
Obtaining Job Information	Obtain the progress of a job with a specified job ID returned after a job request is issued.

Node-related APIs

Table 2-2 Node-related APIs

API	Description
Creating a Node	Create a node in a specified cluster.
Reading a Specified Node	Obtain details about a node with a specified node ID.

API	Description
Reading all Nodes in a Cluster	Obtain details about all nodes in a cluster with a specified cluster ID.
Updating a Specified Node	Update information about a specified node.
Deleting a Node	Delete a specified node
Accepting a Node	Accept a node into a specified cluster.
Resetting a Node	Reset a node in a specified cluster.
Removing a Node	Remove a node from a specified cluster.
Migrating a Node	Migrate a node from a specified cluster to another cluster.

Node Pool-related APIs

Table 2-3 Node pool-related APIs

API	Description
Creating a Node Pool	Create a node pool in a specified cluster.
Reading a Specified Node Pool	Obtain details about a node with a specified node ID.
Listing All Node Pools in a Specified Cluster	Obtain details about all node pools in a cluster with a specified cluster ID.
Updating a Specified Node Pool	Update information about a specified node pool.
Deleting a Node Pool	Delete a specified node pool.

Add-on-related APIs

Table 2-4 Add-on-related APIs

API	Description
Installing an Add-on Instance	Install an add-on by using the add-on template. One or more instances will be created for the installed add-on.
Listing Add-on Templates	Query add-on information.

API	Description
Updating an Add-on Instance	Update an add-on instance.
Deleting an Add-on Instance	Delete an add-on instance.
Reading an Add-on Instance	Obtain details about an add-on instance.
Listing Add-on Instances	List all add-on instances in the cluster.

Quota-related API

Table 2-5 Quota-related API

API	Description
Querying Resource Quotas	Query resource quotas.

Kubernetes APIs

API	Function	URI
Node	Reading a specified node	GET /api/v1/nodes/{name}
	Listing all nodes	GET /api/v1/nodes
	Updating a specified node	PATCH /api/v1/nodes/{name}
Namespace	Creating a namespace	POST /api/v1/namespaces
	Deleting a namespace	DELETE /api/v1/namespaces/{name}
	Querying a specified namespace	GET /api/v1/namespaces/{name}
	Replacing a specified namespace	PUT /api/v1/namespaces/{name}
	Replacing the status of a specified namespace	PUT /api/v1/namespaces/{name}/status
	Replacing the finalize values of a specified namespace	PUT /api/v1/namespaces/{name}/finalize
	Listing namespaces	GET /api/v1/namespaces
	Updating a specified namespace	PATCH /api/v1/namespaces/{name}

API	Function	URI
Resource quotas	Querying resource quotas	GET /api/v1/resourcequotas
	Creating a resource quota	POST /api/v1/namespaces/{namespace}/resourcequotas
	Updating a resource quota	PUT /api/v1/namespaces/{namespace}/resourcequotas/{name}
	Deleting a resource quota	DELETE /api/v1/namespaces/{namespace}/resourcequotas/{name}
Pod	Creating a pod	POST /api/v1/namespaces/{namespace}/pods
	Deleting a pod	DELETE /api/v1/namespaces/{namespace}/pods/{name}
	Deleting all pods	DELETE /api/v1/namespaces/{namespace}/pods
	Reading a specified pod	GET /api/v1/namespaces/{namespace}/pods/{name}
	Replacing a specified pod	PUT /api/v1/namespaces/{namespace}/pods/{name}
	Replacing the status of a specified pod	PUT /api/v1/namespaces/{namespace}/pods/{name}/status
	Listing all pods in a specified namespace	GET /api/v1/namespaces/{namespace}/pods
	Listing pods	GET /api/v1/pods
	Updating a specified pod	PATCH /api/v1/namespaces/{namespace}/pods/{name}
Deployment	Creating a Deployment	POST /apis/apps/v1/namespaces/{namespace}/deployments
	Rolling back a Deployment	PATCH /apis/apps/v1/namespaces/{namespace}/deployments/{name} (for clusters of v1.17 or later) POST /apis/apps/v1beta1/namespaces/{namespace}/deployments/{name}/rollback (for clusters of v1.15 and earlier) POST /apis/extensions/v1beta1/namespaces/{namespace}/deployments/{name}/rollback (for clusters of v1.15 and earlier)
	Deleting a Deployment	DELETE /apis/apps/v1/namespaces/{namespace}/deployments/{name}

API	Function	URI
	Deleting all Deployments	DELETE /apis/apps/v1/namespaces/{namespace}/deployments
	Reading a specified Deployment	GET /apis/apps/v1/namespaces/{namespace}/deployments/{name}
	Reading the status of a specified Deployment	GET /apis/apps/v1/namespaces/{namespace}/deployments/{name}/status
	Reading the scaling operation of a specified Deployment	GET /apis/apps/v1/namespaces/{namespace}/deployments/{name}/scale
	Replacing a specified Deployment	PUT /apis/apps/v1/namespaces/{namespace}/deployments/{name}
	Replacing the status of a specified Deployment	PUT /apis/apps/v1/namespaces/{namespace}/deployments/{name}/status
	Replacing the scaling operation of a specified Deployment	PUT /apis/apps/v1/namespaces/{namespace}/deployments/{name}/scale
	Listing Deployments in a specified namespace	GET /apis/apps/v1/namespaces/{namespace}/deployments
	Listing all Deployments	GET /apis/apps/v1/deployments
	Updating a specified Deployment	PATCH /apis/apps/v1/namespaces/{namespace}/deployments/{name}
	Updating the status of a specified Deployment	PATCH /apis/apps/v1/namespaces/{namespace}/deployments/{name}/status
	Updating the scaling operation of a specified Deployment	PATCH /apis/apps/v1/namespaces/{namespace}/deployments/{name}/scale
StatefulSet	Creating a StatefulSet	POST /apis/apps/v1/namespaces/{namespace}/statefulsets
	Deleting a specified StatefulSet	DELETE /apis/apps/v1/namespaces/{namespace}/statefulsets/{name}
	Deleting all StatefulSets	DELETE /apis/apps/v1/namespaces/{namespace}/statefulsets
	Reading a specified StatefulSet	GET /apis/apps/v1/namespaces/{namespace}/statefulsets/{name}
	Reading the status of a specified StatefulSet	GET /apis/apps/v1/namespaces/{namespace}/statefulsets/{name}/status
	Replacing a specified StatefulSet	PUT /apis/apps/v1/namespaces/{namespace}/statefulsets/{name}

API	Function	URI
	Replacing the status of a specified StatefulSet	PUT /apis/apps/v1/namespaces/{namespace}/statefulsets/{name}/status
	Listing StatefulSets in a specified namespace	GET /apis/apps/v1/namespaces/{namespace}/statefulsets
	Listing all StatefulSets	GET /apis/apps/v1/statefulsets
	Updating a specified StatefulSet	PATCH /apis/apps/v1/namespaces/{namespace}/statefulsets/{name}
	Updating the status of a specified StatefulSet	PATCH /apis/apps/v1/namespaces/{namespace}/statefulsets/{name}/status
Daemon Set	Creating a DaemonSet	POST /apis/apps/v1/namespaces/{namespace}/daemonsets
	Deleting a specified DaemonSet	DELETE /apis/apps/v1/namespaces/{namespace}/daemonsets/{name}
	Deleting all DaemonSets	DELETE /apis/apps/v1/namespaces/{namespace}/daemonsets
	Reading a specified DaemonSet	GET /apis/apps/v1/namespaces/{namespace}/daemonsets/{name}
	Reading the status of a specified DaemonSet	GET /apis/apps/v1/namespaces/{namespace}/daemonsets/{name}/status
	Updating a specified DaemonSet	PATCH /apis/apps/v1/namespaces/{namespace}/daemonsets/{name}
	Updating the status of a specified DaemonSet	PATCH /apis/apps/v1/namespaces/{namespace}/daemonsets/{name}/status
	Listing all DaemonSets	GET /apis/apps/v1/daemonsets
	Listing DaemonSets in a specified namespace	GET /apis/apps/v1/namespaces/{namespace}/daemonsets
	Replacing a specified DaemonSet	PUT /apis/apps/v1/namespaces/{namespace}/daemonsets/{name}
	Replacing the status of a specified DaemonSet	PUT /apis/apps/v1/namespaces/{namespace}/daemonsets/{name}/status
	Job	Creating a job
Deleting a job		DELETE /apis/batch/v1/namespaces/{namespace}/jobs/{name}
Deleting all jobs		DELETE /apis/batch/v1/namespaces/{namespace}/jobs

API	Function	URI
	Reading a specified job	GET /apis/batch/v1/namespaces/{namespace}/jobs/{name}
	Reading the status of a specified job	GET /apis/batch/v1/namespaces/{namespace}/jobs/{name}/status
	Replacing a specified job	PUT /apis/batch/v1/namespaces/{namespace}/jobs/{name}
	Replacing the status of a specified job	PUT /apis/batch/v1/namespaces/{namespace}/jobs/{name}/status
	Listing jobs in a specified namespace	GET /apis/batch/v1/namespaces/{namespace}/jobs
	Listing all jobs	GET /apis/batch/v1/jobs
	Updating the status of a specified job	PATCH /apis/batch/v1/namespaces/{namespace}/jobs/{name}/status
	Updating a specified job	PATCH /apis/batch/v1/namespaces/{namespace}/jobs/{name}
CronJob	Creating a cron job	POST /apis/batch/v1beta1/namespaces/{namespace}/cronjobs
	Deleting a cron job	DELETE /apis/batch/v1beta1/namespaces/{namespace}/cronjobs/{name}
	Deleting all cron jobs	DELETE /apis/batch/v1beta1/namespaces/{namespace}/cronjobs
	Reading a specified cron job	GET /apis/batch/v1beta1/namespaces/{namespace}/cronjobs/{name}
	Reading the status of a specified cron job	GET /apis/batch/v1beta1/namespaces/{namespace}/cronjobs/{name}/status
	Replacing a specified cron job	PUT /apis/batch/v1beta1/namespaces/{namespace}/cronjobs/{name}
	Replacing the status of a specified cron job	PUT /apis/batch/v1beta1/namespaces/{namespace}/cronjobs/{name}/status
	Listing cron jobs under a specified namespace	GET /apis/batch/v1beta1/namespaces/{namespace}/cronjobs
	Listing all cron jobs	GET /apis/batch/v1beta1/cronjobs
	Updating the status of a specified cron job	PATCH /apis/batch/v1beta1/namespaces/{namespace}/cronjobs/{name}/status
	Updating a specified cron job	PATCH /apis/batch/v1beta1/namespaces/{namespace}/cronjobs/{name}

API	Function	URI
ReplicaSet	Listing ReplicaSets	GET /apis/apps/v1/namespaces/{namespace}/replicasets
	Reading a specified ReplicaSet	GET /apis/apps/v1/namespaces/{namespace}/replicasets/{name}
	Listing all ReplicaSets	GET /apis/apps/v1/replicasets
ReplicationController	Creating a ReplicationController	POST /api/v1/namespaces/{namespace}/replicationcontrollers
	Deleting a ReplicationController	DELETE /api/v1/namespaces/{namespace}/replicationcontrollers/{name}
	Deleting all ReplicationControllers	DELETE /api/v1/namespaces/{namespace}/replicationcontrollers
	Reading a ReplicationController under a specified namespace	GET /api/v1/namespaces/{namespace}/replicationcontrollers/{name}
	Replacing a ReplicationController under a specified namespace	PUT /api/v1/namespaces/{namespace}/replicationcontrollers/{name}
	Replacing the status of a ReplicationController under a specified namespace	PUT /api/v1/namespaces/{namespace}/replicationcontrollers/{name}/status
	Listing ReplicationControllers in a specified namespace	GET /api/v1/namespaces/{namespace}/replicationcontrollers
	Listing ReplicationControllers	GET /api/v1/replicationcontrollers
	Updating a specified ReplicationController	PATCH /api/v1/namespaces/{namespace}/replicationcontrollers/{name}
Endpoints	Creating an endpoint	POST /api/v1/namespaces/{namespace}/endpoints
	Deleting an endpoint	DELETE /api/v1/namespaces/{namespace}/endpoints/{name}
	Deleting all endpoints	DELETE /api/v1/namespaces/{namespace}/endpoints
	Querying a specified endpoint	GET /api/v1/namespaces/{namespace}/endpoints/{name}

API	Function	URI
	Replacing a specified endpoint	PUT /api/v1/namespaces/{namespace}/endpoints/{name}
	Listing endpoints	GET /api/v1/endpoints
	Listing endpoints in a specified namespace	GET /api/v1/namespaces/{namespace}/endpoints
	Updating a specified endpoint	PATCH /api/v1/namespaces/{namespace}/endpoints/{name}
Service	Creating a Service	POST /api/v1/namespaces/{namespace}/services
	Deleting a specified Service	DELETE /api/v1/namespaces/{namespace}/services/{name}
	Obtaining a specified Service	GET /api/v1/namespaces/{namespace}/services/{name}
	Replacing a specified Service	PUT /api/v1/namespaces/{namespace}/services/{name}
	Listing Services in a specified namespace	GET /api/v1/namespaces/{namespace}/services
	Listing Services	GET /api/v1/services
	Updating a specified Service	PATCH /api/v1/namespaces/{namespace}/services/{name}
Ingress	Creating an ingress	POST /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses (for clusters of v1.21 and later) POST /apis/networking.k8s.io/v1beta1/namespaces/{namespace}/ingresses (for clusters from v1.15 to v1.21) POST /apis/extensions/v1beta1/namespaces/{namespace}/ingresses (for clusters earlier than v1.15)
	Updating a specified ingress	PATCH /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name} (for clusters of v1.21 and later) PATCH /apis/networking.k8s.io/v1beta1/namespaces/{namespace}/ingresses/{name} (for clusters from v1.15 to v1.21) PATCH /apis/extensions/v1beta1/namespaces/{namespace}/ingresses/{name} (for clusters earlier than v1.15)

API	Function	URI
	Replacing a specified ingress	<p>PUT /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name} (for clusters of v1.21 and later)</p> <p>PUT /apis/networking.k8s.io/v1beta1/namespaces/{namespace}/ingresses/{name} (for clusters from v1.15 to v1.21)</p> <p>PUT /apis/extensions/v1beta1/namespaces/{namespace}/ingresses/{name} (for clusters earlier than v1.15)</p>
	Deleting an ingress	<p>DELETE /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name} (for clusters of v1.21 and later)</p> <p>DELETE /apis/networking.k8s.io/v1beta1/namespaces/{namespace}/ingresses/{name} (for clusters from v1.15 to v1.21)</p> <p>DELETE /apis/extensions/v1beta1/namespaces/{namespace}/ingresses/{name} (for clusters earlier than v1.15)</p>
	Deleting all ingresses	<p>DELETE /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses (for clusters of v1.21 and later)</p> <p>DELETE /apis/networking.k8s.io/v1beta1/namespaces/{namespace}/ingresses (for clusters from v1.15 to v1.21)</p> <p>DELETE /apis/extensions/v1beta1/namespaces/{namespace}/ingresses (for clusters earlier than v1.15)</p>
	Obtaining a specified ingress	<p>GET /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name} (for clusters of v1.21 and later)</p> <p>GET /apis/networking.k8s.io/v1beta1/namespaces/{namespace}/ingresses/{name} (for clusters from v1.15 to v1.21)</p> <p>GET /apis/extensions/v1beta1/namespaces/{namespace}/ingresses/{name} (for clusters earlier than v1.15)</p>

API	Function	URI
	Listing ingresses in a specified namespace	<p>GET /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses (for clusters of v1.21 and later)</p> <p>GET /apis/networking.k8s.io/v1beta1/namespaces/{namespace}/ingresses (for clusters from v1.15 to v1.21)</p> <p>GET /apis/extensions/v1beta1/namespaces/{namespace}/ingresses (for clusters earlier than v1.15)</p>
	Listing ingresses	<p>GET /apis/networking.k8s.io/v1/ingresses (for clusters of v1.21 and later)</p> <p>GET /apis/networking.k8s.io/v1beta1/ingresses (for clusters from v1.15 to v1.21)</p> <p>GET /apis/extensions/v1beta1/ingresses (for clusters earlier than v1.15)</p>
	Obtaining the status of an ingress in a specified namespace	<p>GET /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name}/status (for clusters of v1.21 and later)</p> <p>GET /apis/networking.k8s.io/v1beta1/namespaces/{namespace}/ingresses/{name}/status (for clusters from v1.15 to v1.21)</p> <p>GET /apis/extensions/v1beta1/namespaces/{namespace}/ingresses/{name}/status (for clusters earlier than v1.15)</p>
	Replacing the status of an ingress in a specified namespace	<p>PUT /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name}/status (for clusters of v1.21 and later)</p> <p>PUT /apis/networking.k8s.io/v1beta1/namespaces/{namespace}/ingresses/{name}/status (for clusters from v1.15 to v1.21)</p> <p>PUT /apis/extensions/v1beta1/namespaces/{namespace}/ingresses/{name}/status (for clusters earlier than v1.15)</p>

API	Function	URI
	Updating the status of an ingress in a specified namespace	PATCH /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name}/status (for clusters of v1.21 and later) PATCH /apis/networking.k8s.io/v1beta1/namespaces/{namespace}/ingresses/{name}/status (for clusters from v1.15 to v1.21) PATCH /apis/extensions/v1beta1/namespaces/{namespace}/ingresses/{name}/status (for clusters earlier than v1.15)
Network Policy	Creating a network policy	POST /apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies
	Updating a specified network policy	PATCH /apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies/{name}
	Replacing a specified network policy	PUT /apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies/{name}
	Deleting a specified network policy	DELETE /apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies/{name}
	Deleting network policies in batches	DELETE /apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies
	Reading a specified network policy	GET /apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies/{name}
	Listing network policies in a specified namespace	GET /apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies
	Listing all network policies	GET /apis/networking.k8s.io/v1/networkpolicies
PersistentVolume	Creating a PersistentVolume	POST /api/v1/persistentvolumes
	Deleting a specified PersistentVolume	DELETE /api/v1/persistentvolumes/{name}
	Deleting all PersistentVolumes	DELETE /api/v1/persistentvolumes
	Reading a specified PersistentVolume	GET /api/v1/persistentvolumes/{name}

API	Function	URI
	Replacing a specified PersistentVolume	PUT /api/v1/persistentvolumes/{name}
	Replacing the status of a specified PersistentVolume	PUT /api/v1/persistentvolumes/{name}/status
	Listing all PersistentVolumes	GET /api/v1/persistentvolumes
	Updating a specified PersistentVolume	PATCH /api/v1/persistentvolumes/{name}
PersistentVolumeClaim	Creating a PersistentVolumeClaim	POST /api/v1/namespaces/{namespace}/persistentvolumeclaims
	Deleting a specified PersistentVolumeClaim	DELETE /api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}
	Deleting all PersistentVolume-Claims	DELETE /api/v1/namespaces/{namespace}/persistentvolumeclaims
	Reading a specified PersistentVolumeClaim	GET /api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}
	Replacing a specified PersistentVolumeClaim	PUT /api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}
	Replacing the status of a specified PersistentVolumeClaim	PUT /api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}/status
	Listing PersistentVolume-Claims in a specified namespace	GET /api/v1/namespaces/{namespace}/persistentvolumeclaims
	Listing all PersistentVolume-Claims	GET /api/v1/persistentvolumeclaims
	Updating a specified PersistentVolumeClaim	PATCH /api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}
ConfigMap	Creating a ConfigMap	POST /api/v1/namespaces/{namespace}/configmaps
	Deleting a ConfigMap	DELETE /api/v1/namespaces/{namespace}/configmaps/{name}
	Deleting all ConfigMaps	DELETE /api/v1/namespaces/{namespace}/configmaps

API	Function	URI
	Reading a specified ConfigMap	GET /api/v1/namespaces/{namespace}/configmaps/{name}
	Replacing a specified ConfigMap	PUT /api/v1/namespaces/{namespace}/configmaps/{name}
	Listing ConfigMaps in a specified namespace	GET /api/v1/namespaces/{namespace}/configmaps
	Listing all ConfigMaps	GET /api/v1/configmaps
	Updating a specified ConfigMap	PATCH /api/v1/namespaces/{namespace}/configmaps/{name}
Secret	Creating a secret	POST /api/v1/namespaces/{namespace}/secrets
	Deleting a secret	DELETE /api/v1/namespaces/{namespace}/secrets/{name}
	Deleting all secrets in a specified namespace	DELETE /api/v1/namespaces/{namespace}/secrets
	Reading a specified secret	GET /api/v1/namespaces/{namespace}/secrets/{name}
	Replacing a specified secret	PUT /api/v1/namespaces/{namespace}/secrets/{name}
	Listing the secrets in a namespace	GET /api/v1/namespaces/{namespace}/secrets
	Listing secrets in a cluster	GET /api/v1/secrets
RBAC/ ClusterRole	Creating a ClusterRole	POST /apis/rbac.authorization.k8s.io/v1/clusterroles
	Updating a specified ClusterRole	PATCH /apis/rbac.authorization.k8s.io/v1/clusterroles/{name}
	Replacing a specified ClusterRole	PUT /apis/rbac.authorization.k8s.io/v1/clusterroles/{name}
	Deleting a specified ClusterRole	DELETE /apis/rbac.authorization.k8s.io/v1/clusterroles/{name}
	Deleting ClusterRoles in batches	DELETE /apis/rbac.authorization.k8s.io/v1/clusterroles
	Reading a specified ClusterRole	GET /apis/rbac.authorization.k8s.io/v1/clusterroles/{name}
	Listing ClusterRoles	GET /apis/rbac.authorization.k8s.io/v1/clusterroles

API	Function	URI
RBAC/ ClusterRoleBinding	Creating a ClusterRoleBinding	POST /apis/rbac.authorization.k8s.io/v1/clusterrolebindings
	Updating a specified ClusterRoleBinding	PATCH /apis/rbac.authorization.k8s.io/v1/clusterrolebindings/{name}
	Replacing a specified ClusterRoleBinding	PUT /apis/rbac.authorization.k8s.io/v1/clusterrolebindings/{name}
	Deleting a specified ClusterRoleBinding	DELETE /apis/rbac.authorization.k8s.io/v1/clusterrolebindings/{name}
	Deleting ClusterRoleBindings in batches	DELETE /apis/rbac.authorization.k8s.io/v1/clusterrolebindings
	Reading a specified ClusterRoleBinding	GET /apis/rbac.authorization.k8s.io/v1/clusterrolebindings/{name}
	Listing ClusterRoleBindings	GET /apis/rbac.authorization.k8s.io/v1/clusterrolebindings
RBAC/ Role	Creating a Role	POST /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles
	Updating a specified Role	PATCH /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles/{name}
	Replacing a specified Role	PUT /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles/{name}
	Deleting a specified Role	DELETE /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles/{name}
	Deleting Roles in batches	DELETE /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles
	Reading a specified Role	GET /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles/{name}
	Listing Roles in a specified namespace	GET /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles
	Listing all Roles	GET /apis/rbac.authorization.k8s.io/v1/roles
RBAC/ RoleBinding	Creating a RoleBinding	POST /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings
	Updating a specified RoleBinding	PATCH /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings/{name}
	Replacing a specified RoleBinding	PUT /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings/{name}

API	Function	URI
	Deleting a specified RoleBinding	DELETE /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings/{name}
	Deleting RoleBindings in batches	DELETE /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings
	Reading a specified RoleBinding	GET /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings/{name}
	Listing RoleBindings in a specified namespace	GET /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings
	Listing all RoleBindings	GET /apis/rbac.authorization.k8s.io/v1/rolebindings
API groups	Listing APIVersions	GET /api
	Listing APIGroups	GET /apis
	listing APIResources of GroupVersion apiregistration.k8s.io/v1beta1	GET /apis/apiregistration.k8s.io/v1beta1
	listing APIResources of GroupVersion extensions/v1beta1	GET /apis/extensions/v1beta1
	listing APIResources of GroupVersion apps/v1&apps/v1beta1	GET /apis/apps/v1 (for clusters later than v1.15) GET /apis/apps/v1beta1 (for clusters of v1.15 and earlier)
	listing APIResources of GroupVersion authentication.k8s.io/v1	GET /apis/authentication.k8s.io/v1
	listing APIResources of GroupVersion authentication.k8s.io/v1beta1	GET /apis/authentication.k8s.io/v1beta1
	listing APIResources of GroupVersion authorization.k8s.io/v1	GET /apis/authorization.k8s.io/v1
	listing APIResources of GroupVersion authorization.k8s.io/v1beta1	GET /apis/authorization.k8s.io/v1beta1

API	Function	URI
	listing APIResources of GroupVersion autoscaling/v1	GET /apis/autoscaling/v1
	listing APIResources of GroupVersion batch/v1	GET /apis/batch/v1
	listing APIResources of GroupVersion certificates.k8s.io/v1beta1	GET /apis/certificates.k8s.io/v1beta1
	listing APIResources of GroupVersion networking.k8s.io/v1	GET /apis/networking.k8s.io/v1
	listing APIResources of GroupVersion policy/v1beta1	GET /apis/policy/v1beta1
	listing APIResources of GroupVersion rbac.authorization.k8s.io/v1beta1	GET /apis/rbac.authorization.k8s.io/v1beta1
	listing APIResources of GroupVersion storage.k8s.io/v1	GET /apis/storage.k8s.io/v1
	listing APIResources of GroupVersion storage.k8s.io/v1beta1	GET /apis/storage.k8s.io/v1beta1
	listing APIResources of GroupVersion apiextensions.k8s.io/v1beta1	GET /apis/apiextensions.k8s.io/v1beta1
	listing APIResources of GroupVersion v1	GET /api/v1
Event	Reading events	GET /api/v1/events
	Listing events in a specified namespace	GET /api/v1/namespaces/{namespace}/events

3 Calling APIs

3.1 Making an API Request

This section describes the structure of a REST API request, and uses the IAM API for **obtaining a user token** as an example to demonstrate how to call an API. The obtained token can then be used to authenticate the calling of other APIs.

Request URI

A request URI is in the following format:

{URI-scheme}://{Endpoint}/{resource-path}?{query-string}

Although a request URI is included in the request header, most programming languages or frameworks require the request URI to be transmitted separately.

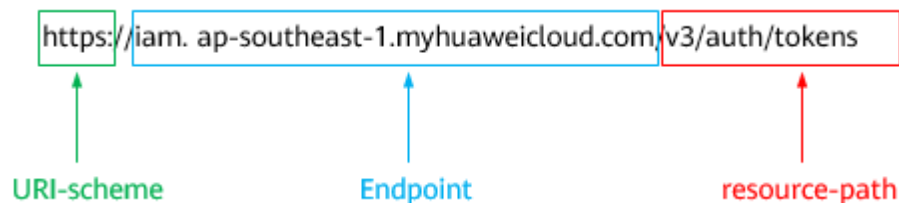
Table 3-1 URI parameters

Parameter	Description
URI-scheme	Protocol used to transmit requests. All APIs use HTTPS.
Endpoint	Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from Regions and Endpoints . For example, the endpoint of IAM in region CN-Hong Kong is iam.ap-southeast-1.myhuaweicloud.com .
resource-path	Access path of an API for performing an operation. Obtain the path from the URI of an API. For example, the resource-path of the API used to obtain a user token is /v3/auth/tokens .
query-string	Query parameter, which is optional. Ensure that a question mark (?) is included before each query parameter that is in the format of <i>Parameter name=Parameter value</i> . For example, ?limit=10 indicates that a maximum of 10 data records will be displayed.

For example, to obtain an IAM token in the **CN-Hong Kong** region, obtain the endpoint of IAM (**iam.ap-southeast-1.myhuaweicloud.com**) for this region and the **resource-path (/v3/auth/tokens)** in the URI of the API used to **obtain a user token**. Then, construct the URI as follows:

```
https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
```

Figure 3-1 Example URI



NOTE

To simplify the URI display in this document, each API is provided only with a **resource-path** and a request method. The **URI-scheme** of all APIs is **HTTPS**, and the endpoints of all APIs in the same region are identical.

Request Methods

The HTTP protocol defines the following request methods that can be used to send a request to the server.

Table 3-2 HTTP methods

Method	Description
GET	Requests the server to return specified resources.
PUT	Requests the server to update specified resources.
POST	Requests the server to add resources or perform special operations.
DELETE	Requests the server to delete specified resources, for example, an object.
HEAD	Same as GET except that the server must return only the response header.
PATCH	Requests the server to update partial content of a specified resource. If the resource does not exist, a new resource will be created.

For example, in the case of the API used to **obtain a user token**, the request method is **POST**. The request is as follows:

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
```

Request Header

You can also add additional header fields to a request, such as the fields required by a specified URI or HTTP method. For example, to request for the authentication information, add **Content-Type**, which specifies the request body type.

Common request header fields are as follows.

Table 3-3 Common request header fields

Parameter	Description	Mandatory	Example Value
Host	Specifies the server domain name and port number of the resources being requested. The value can be obtained from the URL of the service API. The value is in the format of <i>Hostname:Port number</i> . If the port number is not specified, the default port is used. The default port number for https is 443 .	No This field is mandatory for AK/SK authentication.	code.test.com or code.test.com:443
Content-Type	Specifies the type (or format) of the message body. The default value application/json is recommended. Other values of this field will be provided for specific APIs if any.	Yes	application/json
Content-Length	Specifies the length of the request body. The unit is byte.	No	3495

Parameter	Description	Mandatory	Example Value
X-Project-Id	Specifies the project ID. Obtain the project ID by following the instructions in Obtaining a Project ID .	No This field is mandatory for requests that use AK/SK authentication in the Dedicated Cloud (DeC) scenario or multi-project scenario.	e9993fc787d94b6c886cb aa340f9c0f4
X-Auth-Token	Specifies the user token. It is a response to the API for obtaining a user token (This is the only API that does not require authentication). After the request is processed, the value of X-Subject-Token in the response header is the token value.	No This field is mandatory for token authentication.	The following is part of an example token: MIIPAgYJKoZIhvcNAQc- Co...ggg1BBIINPXsidG9rZ

 **NOTE**

In addition to supporting authentication using tokens, APIs support authentication using AK/SK, which uses SDKs to sign a request. During the signature, the **Authorization** (signature authentication) and **X-Sdk-Date** (time when a request is sent) headers are automatically added in the request.

For more details, see "Authentication Using AK/SK" in [Authentication](#).

The API used to [obtain a user token](#) does not require authentication. Therefore, only the **Content-Type** field needs to be added to requests for calling the API. An example of such requests is as follows:

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
```

(Optional) Request Body

This part is optional. The body of a request is often sent in a structured format as specified in the **Content-Type** header field. The request body transfers content except the request header.

The request body varies between APIs. Some APIs do not require the request body, such as the APIs requested using the GET and DELETE methods.

In the case of the API used to [obtain a user token](#), the request parameters and parameter description can be obtained from the API request. The following provides an example request with a body included. Replace ***username***, ***domainname***, *********** (login password), and ***xxxxxxxxxxxxxxxxxxxx*** (project name) with the actual values. Obtain a project name from [Regions and Endpoints](#).

NOTE

The **scope** parameter specifies where a token takes effect. You can set **scope** to an account or a project under an account. In the following example, the token takes effect only for the resources in a specified project. For more information about this API, see [Obtaining a User Token](#).

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
```

```
Content-Type: application/json
```

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username",
          "password": "*****",
          "domain": {
            "name": "domainname"
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "xxxxxxxxxxxxxxxxxxxx"
      }
    }
  }
}
```

If all data required for the API request is available, you can send the request to call the API through [curl](#), [Postman](#), or coding. In the response to the API used to obtain a user token, **x-subject-token** is the desired user token. This token can then be used to authenticate the calling of other APIs.

3.2 Authentication

Requests for calling an API can be authenticated using either of the following methods:

- Token authentication: Requests are authenticated using tokens.
- AK/SK authentication: Requests are encrypted using an AK/SK. AK/SK-based authentication is recommended because it is more secure than token-based authentication.

Token-based Authentication

NOTE

The validity period of a token is 24 hours. When using a token for authentication, cache it to prevent frequently calling the IAM API used to obtain a user token.

A token specifies temporary permissions in a computer system. During API authentication using a token, the token is added to requests to get permissions for calling the API.

You can obtain a token by calling the [Obtaining a User Token](#) API. When you call the API, set **auth.scope** in the request body to **project**.

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username",
          "password": "*****",
          "domain": {
            "name": "domainname"
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "xxxxxxx"
      }
    }
  }
}
```

After a token is obtained, the **X-Auth-Token** header field must be added to requests to specify the token when calling other APIs. For example, if the token is **ABCDEFJ....**, **X-Auth-Token: ABCDEFJ....** can be added to a request as follows:

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/projects
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

AK/SK Authentication

NOTE

AK/SK authentication supports API requests with a body not larger than 12 MB. For API requests with a larger body, token authentication is recommended.

In AK/SK authentication, AK/SK is used to sign requests and the signature is then added to the requests for authentication.

- AK: access key ID, which is a unique identifier used in conjunction with a secret access key to sign requests cryptographically.
- SK: secret access key, which is used in conjunction with an AK to sign requests cryptographically. It identifies a request sender and prevents the request from being modified.

In AK/SK authentication, you can use an AK/SK to sign requests based on the signature algorithm or using the signing SDK. For details about how to sign requests and use the signing SDK, see [API Request Signing Guide](#).

 **NOTE**

The signing SDK is only used for signing requests and is different from the SDKs provided by services.

3.3 Response

Status Code

After sending a request, you will receive a response, including a status code, response header, and response body.

A status code is a group of digits, ranging from 1xx to 5xx. It indicates the status of a request. For more information, see [Status Code](#).

For example, if status code **201** is returned for calling the API used to [obtain a user token](#), the request is successful.

Response Header

Similar to a request, a response also has a header, for example, **Content-Type**.

[Figure 3-2](#) shows the response header fields for the API used to [obtain a user token](#). The **x-subject-token** header field is the desired user token. This token can then be used to authenticate the calling of other APIs.

Figure 3-2 Header fields of the response to the request for obtaining a user token

```
connection -- keep-alive
content-type -- application/json
date -- Tue, 12 Feb 2019 06:52:13 GMT
server -- Web Server
strict-transport-security -- max-age=31536000; includeSubdomains;
transfer-encoding -- chunked
via -- proxy A
x-content-type-options -- nosniff
x-download-options -- noopen
x-frame-options -- SAMEORIGIN
x-iam-trace-id -- 218d45ab-d674-4995-af3a-2d0255ba41b5
x-subject-token -- [redacted]
x-xss-protection -- 1; mode=block
```

Response Body

The body of a response is often returned in structured format as specified in the **Content-Type** header field. The response body transfers content except the response header.

The following is part of the response body for the API used to **obtain a user token**.

```
{
  "token": {
    "expires_at": "2019-02-13T06:52:13.855000Z",
    "methods": [
      "password"
    ],
    "catalog": [
      {
        "endpoints": [
          {
            "region_id": "ap-southeast-1",
            .....

```

If an error occurs during API calling, an error code and a message will be displayed. The following shows an error response body.

```
{
  "error_msg": "The format of message is error",
  "error_code": "AS.0001"
}
```

In the response body, **error_code** is an error code, and **error_msg** provides information about the error.

4 APIs

4.1 API URL

- The URL format for cluster, node, node pool, add-on, and quota management is **https://Endpoint/uri**. *uri* indicates the resource path, that is, the API access path.
- The URL format for Kubernetes APIs, storage management, and add-on management is **https://{clusterid}.Endpoint/uri**. In the URL, *{clusterid}* indicates the cluster ID, and *uri* indicates the resource path, that is, the path for API access.

 NOTE

- The format of the URL called by the add-on management APIs is **https://{clusterid}.Endpoint/uri**. However, *{clusterid}* is used only for the domain name and is not verified or used by the APIs. Set *{clusterid}* in the query or body. For details about *{clusterid}*, see the add-on management sections.
- *{clusterid}* is required for Kubernetes APIs and storage management, which indicates the cluster that needs to be accessed by calling the API.

Table 4-1 URL parameters

Parameter	Description
{clusterid}	Cluster ID. After a cluster is created, call the API for obtaining a cluster in a specified project to obtain the cluster ID.
Endpoint	URL that is the entry point for a web service. You can obtain it from Endpoints .
uri	Access path of an API for performing an operation. Obtain the path from the URI of an API. For example, the resource-path of the API used to obtain a user token is v3/auth/tokens .

4.2 Cluster Management

4.2.1 Creating a Cluster

Function

This API is used to create an empty cluster, which has only master nodes but no worker nodes. After creating a cluster by calling this API, you can add nodes by [creating nodes](#).

NOTE

- The URL for cluster management is in the format of **https://Endpoint/uri**, in which *uri* indicates the resource path, that is, the path for API access.
- By default, ICAgent is not installed when you call this API to create a cluster. If you need to install ICAgent, add "cluster.install.addons.external/install":[{"addonTemplateName":"icagent"}]" to **annotations** in the request body. ICAgent will be automatically installed during cluster creation. ICAgent is an O&M data collection agent used by Application Performance Management (APM). It runs on each server to collect data from probes in real time. ICAgent is the prerequisite for achieving application O&M. If ICAgent is not installed, the application O&M functions cannot be used.

Constraints

Before calling the CCE API to create a cluster, ensure that the following conditions are met:

- A VPC is available. Otherwise, the cluster cannot be created. If a VPC is available, you do not need to create a new one. A VPC provides an isolated, configurable, and manageable virtual network environment for CCE clusters. If no VPC is available, you need to create one first. For details, see [Creating a VPC](#).
- CIDR blocks have been properly configured for containers and Services. They are not editable after cluster creation, unless you create a new cluster.
- An agency has been correctly created and is not deleted. If the agency verification fails, the cluster fails to be created. You can log in to the CCE console. If no agency is created, the system prompts you to create one. If an agency has been created, no message is displayed.
- By default, an account can create up to five clusters in each region. If you need to create more clusters, you can submit an application to increase the quota. For details, see [How Do I Increase My Quota?](#).

Calling Method

For details, see [Calling APIs](#).

URI

POST /api/v3/projects/{project_id}/clusters

Table 4-2 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-3 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Table 4-4 Request body parameters

Parameter	Mandatory	Type	Description
kind	Yes	String	API type. The value is fixed at Cluster or cluster and cannot be changed.
apiVersion	Yes	String	API version. The value is fixed at v3 and cannot be changed.
metadata	Yes	ClusterMetadata object	Basic information about a cluster. Metadata is a collection of attributes.
spec	Yes	ClusterSpec object	Detailed description of the cluster. CCE creates or updates objects by defining or updating spec .
status	No	ClusterStatus object	Cluster status and job ID of the cluster creation job.

Table 4-5 ClusterMetadata

Parameter	Mandatory	Type	Description
name	Yes	String	Cluster name. Enter 4 to 128 characters, starting with a lowercase letter and not ending with a hyphen (-). Only lowercase letters, digits, and hyphens (-) are allowed.
uid	No	String	Cluster ID, which uniquely identifies a cluster. This ID is automatically generated after a cluster is created. Only the automatically generated ID will take effect. When you create a yearly/monthly cluster, no cluster ID will be returned in the response body.
alias	No	String	Alias of a cluster name displayed on the CCE console, and the name can be changed. Enter 4 to 128 characters, starting with a lowercase letter and not ending with a hyphen (-). Only lowercase letters, digits, and hyphens (-) are allowed. A cluster alias must be unique. In the request body for creating or updating a cluster, if the cluster alias is not specified or set to null, the cluster name will be used as the cluster alias. In the response body for obtaining a cluster, the cluster alias is returned. If it is not configured, the cluster name will be returned.

Parameter	Mandatory	Type	Description
annotations	No	Map<String,String>	<p>Cluster annotations, in the format of key-value pairs.</p> <pre>"annotations": { "key1": "value1", "key2": "value2" }</pre> <p>NOTE</p> <ul style="list-style-type: none"> • annotations: Does not label or select objects. The metadata in annotations may be small or large, structured or unstructured, and may include characters that are not allowed in labels. • This field is not stored in the database and is used only to specify the add-ons to be installed in the cluster. • Install ICAgent during cluster creation by adding the key-value pair "cluster.install.addons.external / install":{"addonTemplateName":"icagent"}".
labels	No	Map<String,String>	<p>Cluster labels, in the format of key-value pairs.</p> <p>NOTE</p> <p>The value of this field is automatically generated by the system and is used by the frontend to identify the features supported by the cluster during the upgrade. Customized values are invalid.</p>
creationTimestamp	No	String	Time when the cluster was created.
updateTimestamp	No	String	Time when the cluster was updated.

Table 4-6 ClusterSpec

Parameter	Mandatory	Type	Description
category	No	String	<p>Cluster type. Options:</p> <ul style="list-style-type: none"> • CCE: CCE cluster CCE cluster supports hybrid deployment of VMs and BMSs, and heterogeneous nodes such as GPU and NPU nodes, allowing you to run your containers in a secure and stable container runtime environment based on a high-performance network model. • Turbo: CCE Turbo cluster One-stop, cost-effective CCE Turbo clusters run on the cloud native 2.0 infrastructure featuring hardware-software synergy for lossless networking, high security and reliability, and intelligent scheduling.
type	No	String	<p>Master node architecture:</p> <ul style="list-style-type: none"> • VirtualMachine: x86 • ARM64: Arm-based Kunpeng

Parameter	Mandatory	Type	Description
flavor	Yes	String	<p>Cluster specifications. Specifications of clusters 1.15 or later versions can be changed after they are created. For details, see Modifying Cluster Specifications. Options:</p> <ul style="list-style-type: none"> • cce.s1.small: a small-scale CCE cluster with one master node and a maximum of 50 worker nodes • cce.s1.medium: a medium-scale CCE cluster with one master node and a maximum of 200 worker nodes • cce.s2.small: a small-scale CCE cluster with three master nodes and a maximum of 50 worker nodes • cce.s2.medium: a medium-scale CCE cluster with three master nodes and a maximum of 200 worker nodes • cce.s2.large: a large-scale CCE cluster with three master nodes and a maximum of 1,000 worker nodes • cce.s2.xlarge: an ultra-large-scale CCE cluster with three master nodes and a maximum of 2,000 worker nodes

Parameter	Mandatory	Type	Description
			<p>NOTE The fields in the parameters are described as follows:</p> <ul style="list-style-type: none"> • s1: specifies a cluster with one master node. If the master node is faulty, the cluster will become unavailable, but running workloads in the cluster are not affected. • s2: specifies an HA cluster with three master nodes. If one of the master nodes is faulty, the cluster is still available. • dec: specifies a DeC CCE cluster. For example, cce.dec.s1.small specifies a small-scale, DeC CCE cluster with one master node and a maximum of 50 worker nodes. • small: specifies that a cluster can manage a maximum of 50 worker nodes. • medium: specifies that a cluster can manage a maximum of 200 worker nodes. • large: specifies that a cluster can manage a maximum of 1,000 worker nodes. • xlarge: specifies that a cluster can manage a maximum of 2,000 worker nodes.

Parameter	Mandatory	Type	Description
version	No	String	<p>Cluster version, which mirrors the baseline version of the Kubernetes community. The latest version is recommended.</p> <p>You can create clusters of two latest versions on the CCE console. To learn which cluster versions are available, log in to the CCE console, create a cluster, and check the Cluster Version parameter.</p> <p>You can call APIs to create clusters of other versions. However, these cluster versions will be gradually terminated. For details about the support policy, see the CCE announcement.</p> <p>NOTE</p> <ul style="list-style-type: none"> • If not specified, a cluster of the latest version will be created. • If a baseline cluster version is specified but the R version is not specified, a cluster of the latest R version will be created by default. It is a good practice not to specify the R version. • CCE Turbo clusters of v1.19 or later are commercially available.

Parameter	Mandatory	Type	Description
platformVersion	No	String	<p>CCE cluster platform version, indicating the internal version under the cluster version (version). Platform versions are used to trace iterations in a major cluster version. They are unique within a major cluster version and recounted when the major cluster version changes. This parameter cannot be customized. When you create a cluster, the latest corresponding platform version is automatically selected.</p> <p>The format of platformVersion is cce.X.Y.</p> <ul style="list-style-type: none"> • X: internal feature version, indicating changes in features, patches, or OS support in the cluster version. The value starts from 1 and increases monotonically. • Y: patch version of an internal feature version. It is used only for software package update after the feature version is brought online. No other modification is involved. The value starts from 0 and increases monotonically.
description	No	String	<p>Cluster description, for example, which purpose the cluster is intended to serve. By default, this field is left unspecified. To modify cluster description after the cluster is created, call the API for updating cluster information or go to the cluster details page on the CCE console. Only UTF-8 encoding is supported.</p>

Parameter	Mandatory	Type	Description
customSan	No	Array of strings	<p>Custom SAN field in the server certificate of the cluster API server, which must comply with the SSL and X509 format specifications.</p> <ol style="list-style-type: none"> 1. Duplicate names are not allowed. 2. Must comply with the IP address and domain name formats. <p>Example: SAN 1: DNS Name=example.com SAN 2: DNS Name=www.example.com SAN 3: DNS Name=example.net SAN 4: IP Address=93.184.216.34</p>
ipv6enable	No	Boolean	Whether the cluster supports IPv6 addresses. This field is supported in clusters of v1.15 and later versions.
hostNetwork	Yes	HostNetwork object	Node networking parameters, including VPC and subnet ID. This field is mandatory because nodes in a cluster communicate with each other by using a VPC.
containerNetwork	Yes	ContainerNetwork object	Container networking parameters, including the container network model and container CIDR block.
eniNetwork	No	EniNetwork object	Configuration of the Cloud Native Network 2.0 model. Specify this field when creating a CCE Turbo cluster.
serviceNetwork	No	ServiceNetwork object	Service CIDR block, including IPv4 CIDR blocks.
authentication	No	Authentication object	Configurations of the cluster authentication mode.
billingMode	No	Integer	<p>Billing mode of a cluster.</p> <ul style="list-style-type: none"> • 0: pay-per-use • 1: yearly/monthly <p>Defaults to pay-per-use.</p>
masters	No	Array of MasterSpec objects	Advanced configurations of master nodes

Parameter	Mandatory	Type	Description
kubernetesSvcIpRange	No	String	Service CIDR blocks for Kubernetes cluster IPs. This field is available only for clusters of v1.11.7 and later. If this parameter is not specified during cluster creation, the default value 10.247.0.0/16 will be used. This parameter is deprecated. Use serviceNetwork instead. The new field contains the IPv4 CIDR blocks.
clusterTags	No	Array of ResourceTag objects	Cluster resource tags.
kubeProxyMode	No	String	Service forwarding mode. Options: <ul style="list-style-type: none"> • iptables: Traditional kube-proxy uses iptables rules to implement Service load balancing. In this mode, too many iptables rules will be generated when many Services are deployed. In addition, non-incremental updates will cause latency and even tangible performance issues in the case of service traffic spikes. • ipvs: Optimized kube-proxy mode with higher throughput and faster speed. This mode supports incremental updates and can keep connections uninterrupted during Service updates. It is suitable for large-sized clusters. <p>NOTE iptables is used by default.</p>
az	No	String	AZ. This field is returned only for a query. For details about AZs supported by CCE, see Regions and Endpoints .

Parameter	Mandatory	Type	Description
extendParam	No	ClusterExtendedParam object	Extended field to decide whether the cluster will span across AZs or belong to a specified enterprise project, or whether a dedicated CCE cluster is to be created.
supportIstio	No	Boolean	Whether Istio is supported.
configurationsOverride	No	Array of PackageConfiguration objects	Cluster default component configuration override. If you specify a component or parameter that is not supported, the configuration item will be ignored. For details about the supported components and their parameters, see Cluster Configuration Management .

Table 4-7 HostNetwork

Parameter	Mandatory	Type	Description
vpc	Yes	String	ID of the VPC used to create a master node. You can obtain it in either of the following ways: <ul style="list-style-type: none"> Method 1: Log in to the VPC console and view the VPC ID in the VPC details page. Method 2: Use the VPC API to obtain VPC IDs. For details, see Querying VPCs .

Parameter	Mandatory	Type	Description
subnet	Yes	String	<p>Network ID of the subnet used to create a master node. You can obtain it in either of the following ways:</p> <ul style="list-style-type: none"> • Method 1: Log in to the VPC console and click the target subnet on the Subnets page. You can view the network ID on the displayed page. • Method 2: Use the VPC API to obtain subnets. <p>For details, see Querying Subnets.</p>
SecurityGroup	No	String	<p>Default worker node security group ID of the cluster. If specified, the cluster will be bound to the target security group. Otherwise, the system will automatically create a default worker node security group for you. The default worker node security group needs to allow access from certain ports to ensure normal communications. For details, see How Do I Harden the Automatically Created Security Group Rules for CCE Cluster Nodes.</p>

Table 4-8 ContainerNetwork

Parameter	Mandatory	Type	Description
mode	Yes	String	<p>Container network model. Select one of the following possible values:</p> <ul style="list-style-type: none"> • overlay_l2: an overlay_l2 network (container tunnel network) built for containers by using OpenVSwitch (OVS). • vpc-router: an underlay_l2 network built for containers by using IPvlan and custom VPC routes. • eni: Cloud Native Network 2.0. This model has integrated cloud native elastic network interfaces (ENIs), uses VPC CIDR blocks to allocate container IP addresses, and allows direct traffic distribution to containers through a load balancer for high performance. Use this model when creating a CCE Turbo cluster.
cidr	No	String	<p>Container CIDR block. Recommended: 10.0.0.0/12-19, 172.16.0.0/16-19, or 192.168.0.0/16-19. If the selected CIDR block conflicts with existing ones, an error will be reported.</p> <p>Not editable after the cluster is created. (deprecated. A specified cidrs will make cidr invalid.)</p>

Parameter	Mandatory	Type	Description
cidrs	No	Array of ContainerCIDR objects	List of container CIDR blocks. In clusters of v1.21 or later, the cidrs field is used. When the cluster network type is vpc-router , you can configure a maximum of 20 container CIDR blocks. In cluster versions earlier than v1.21, if the cidrs field is used, the first CIDR element in the array is used as the container CIDR block. The configuration cannot be modified after the cluster is created.

Table 4-9 ContainerCIDR

Parameter	Mandatory	Type	Description
cidr	Yes	String	Container CIDR block. Recommended: 10.0.0.0/12-19, 172.16.0.0/16-19, and 192.168.0.0/16-19

Table 4-10 EniNetwork

Parameter	Mandatory	Type	Description
eniSubnetId	Yes	String	IPv4 subnet ID of ENI subnet. (IPv6 is not supported and is being discarded.) You can obtain it in either of the following ways: <ul style="list-style-type: none"> Method 1: Log in to the VPC console and click the target subnet on the Subnets page. You can view the IPv4 subnet ID on the displayed page. Method 2: Use the VPC API to obtain subnet IDs. For details, see Querying Subnets .
eniSubnetCIDR	No	String	ENI subnet CIDR (being discarded)

Parameter	Mandatory	Type	Description
subnets	Yes	Array of NetworkSubnet objects	List of IPv4 subnet IDs

Table 4-11 NetworkSubnet

Parameter	Mandatory	Type	Description
subnetID	Yes	String	<p>IPv4 subnet ID of the subnet for creating master nodes. IPv6 is not supported. You can obtain it in either of the following ways:</p> <ul style="list-style-type: none"> Method 1: Log in to the VPC console and click the target subnet on the Subnets page. You can view the IPv4 subnet ID on the displayed page. Method 2: Use the VPC API to obtain subnets. <p>For details, see Querying Subnets.</p>

Table 4-12 ServiceNetwork

Parameter	Mandatory	Type	Description
IPv4CIDR	No	String	Value range of the Kubernetes clusterIP IPv4 CIDR blocks. If this parameter is not specified during cluster creation, the default value 10.247.0.0/16 will be used.

Table 4-13 Authentication

Parameter	Mandatory	Type	Description
mode	No	String	Cluster authentication mode. <ul style="list-style-type: none"> Clusters of Kubernetes v1.11 or earlier support x509, rbac, and authenticating_proxy. Defaults to x509. Clusters of Kubernetes v1.13 or later support rbac and authenticating_proxy. Defaults to rbac.
authenticatingProxy	No	AuthenticatingProxy object	Configuration related to the <code>authenticating_proxy</code> mode. This field is mandatory when the authentication mode is authenticating_proxy .

Table 4-14 AuthenticatingProxy

Parameter	Mandatory	Type	Description
ca	No	String	X509 CA certificate (Base64-encoded) configured in <code>authenticating_proxy</code> mode. This field is mandatory when the cluster authentication mode is authenticating_proxy . Maximum size: 1 MB
cert	No	String	Client certificate issued by the X509 CA certificate configured in <code>authenticating_proxy</code> mode, which is used for authentication from kube-apiserver to the extended API server. (The value must be Base64-encoded.) This field is mandatory when the cluster authentication mode is authenticating_proxy .

Parameter	Mandatory	Type	Description
privateKey	No	String	Private key of the client certificate issued by the X509 CA certificate configured in <code>authenticating_proxy</code> mode, which is used for authentication from kube-apiserver to the extended API server. The private key used by the Kubernetes cluster does not support password encryption. Use an unencrypted private key. (The value must be Base64-encoded.) This field is mandatory when the cluster authentication mode is authenticating_proxy .

Table 4-15 MasterSpec

Parameter	Mandatory	Type	Description
availabilityZone	No	String	AZ

Table 4-16 ResourceTag

Parameter	Mandatory	Type	Description
key	No	String	Key. <ul style="list-style-type: none"> • Cannot be null. Max characters: 128. • Use letters, digits, and spaces in UTF-8 format. • Can contain the following special characters: <code>._:/=+@</code>. • Cannot start with <code>_sys_</code>.

Parameter	Mandatory	Type	Description
value	No	String	<p>Value.</p> <ul style="list-style-type: none"> • Can be null but not the default. Max characters: 255. • Use letters, digits, and spaces in UTF-8 format. • Can contain the following special characters: <code>./=+@</code>.

Table 4-17 ClusterExtendParam

Parameter	Mandatory	Type	Description
clusterAZ	No	String	<p>AZ of master nodes in a cluster.</p> <p>For details about AZs supported by CCE, see Regions and Endpoints.</p> <ul style="list-style-type: none"> • multi_az: (Optional) The cluster will span across AZs. Multiple AZs can be configured only when a cluster with multiple master nodes is used. • <i>AZ of the dedicated cloud computing pool</i>: The cluster will be deployed in the DeC AZ. It is mandatory for DeC CCE clusters.

Parameter	Mandatory	Type	Description
dssMasterVolumes	No	String	<p>Whether the system and data disks of a master node use dedicated distributed storage. If this parameter is omitted or left unspecified, EVS disks are used by default.</p> <p>This parameter is mandatory for dedicated CCE clusters. It is in the following format: <code><rootVol.dssPoolID>.<rootVol.volType>;<dataVol.dssPoolID>.<dataVol.volType></code></p> <p>Field description:</p> <ul style="list-style-type: none"> • rootVol is the system disk. dataVol is the data disk. • dssPoolID indicates the ID of the DSS storage pool. • volType indicates the storage volume type of the DSS storage pool, such as SAS and SSD. <p>Example: c950ee97-587c-4f24-8a74-3367e3da570f.sas;6edbc2f4-1507-44f8-ac0d-eed1d2608d38.ssd</p> <p>NOTE This field cannot be configured for non-dedicated CCE clusters.</p>
enterpriseProjectId	No	String	<p>ID of the enterprise project that a cluster belongs to.</p> <p>NOTE</p> <ul style="list-style-type: none"> • An enterprise project can be configured only after the enterprise project function is enabled. • The enterprise project to which the cluster belongs must be the same as that to which other cloud service resources associated with the cluster belong.

Parameter	Mandatory	Type	Description
kubeProxyMode	No	String	<p>Service forwarding mode. Two modes are available:</p> <ul style="list-style-type: none"> • iptables: Traditional kube-proxy uses iptables rules to implement Service load balancing. In this mode, too many iptables rules will be generated when many Services are deployed. In addition, non-incremental updates will cause latency and even tangible performance issues in the case of service traffic spikes. • ipvs: Optimized kube-proxy mode with higher throughput and faster speed. This mode supports incremental updates and can keep connections uninterrupted during Service updates. It is suitable for large-sized clusters. <p>NOTE This parameter has been deprecated. If this parameter and kubeProxyMode in ClusterSpec are specified at the same time, the latter is used.</p>
clusterExternalIP	No	String	EIP of the master node

Parameter	Mandatory	Type	Description
alpha.cce/ fixPoolMask	No	String	<p>Number of mask bits of the fixed IP address pool of the container network model. This field is supported only for the VPC network model (vpc-router).</p> <p>This parameter determines the number of container IP addresses that can be allocated to a node. The maximum number of pods that can be created on a node is decided by this parameter and maxPods set during node creation.</p> <p>For details, see Maximum Number of Pods That Can Be Created on a Node.</p> <p>For integer characters, the value ranges from 24 to 28.</p>
decMasterFlavor	No	String	Specifications of the master node in the dedicated hybrid cluster.
dockerUmaskMode	No	String	Default UmaskMode configuration of Docker in a cluster. The value can be secure or normal . If this parameter is not specified, normal is used by default.

Parameter	Mandatory	Type	Description
kubernetes.io/cpuManagerPolicy	No	String	<p>Cluster CPU management policy. The value can be none (or null) or static. The default value is none (or null).</p> <ul style="list-style-type: none"> • none or null: disables pods from exclusively occupying CPUs. Select this option if you want a large pool of shareable CPU cores. • static: enables pods to exclusively occupy CPUs. Select this option if your workload is sensitive to CPU cache and scheduling latency. In a CCE Turbo cluster, this setting is valid only for nodes where common containers, not Kata containers, run.
orderId	No	String	<p>Order ID. This parameter is returned in the response when the cluster is billed on a yearly/monthly basis with auto payment enabled (only in creation scenarios).</p>
periodType	No	String	<ul style="list-style-type: none"> • month: The unit is month. • year: The unit is year. <p>NOTE Request parameter, which is valid and mandatory when billingMode is set to 1 (yearly/monthly billing). Response parameter, which is returned only when a yearly/monthly cluster is created.</p>

Parameter	Mandatory	Type	Description
periodNum	No	Integer	Subscription duration. The value can be: <ul style="list-style-type: none"> • If periodType is month, the value ranges from 1 to 9. • If periodType is year, the value ranges from 1 to 3. NOTE Request parameter, which is valid and mandatory when billingMode is set to 1 . Response parameter, which is returned only when a yearly/monthly cluster is created.
isAutoRenew	No	String	Whether auto renewal is enabled. <ul style="list-style-type: none"> • true: Auto renewal is enabled. • false: Auto renewal is not enabled. NOTE This field is valid when billingMode is set to 1 . If not specified, auto renewal is not enabled.
isAutoPay	No	String	Whether to deduct fees automatically. <ul style="list-style-type: none"> • true: Enable automatic fee deduction. • false: Do not enable automatic fee deduction. NOTE This field is valid when billingMode is set to 1 . If not specified, auto fee deduction is not enabled.
upgradefrom	No	String	Records of how the cluster is upgraded to the current version

Table 4-18 PackageConfiguration

Parameter	Mandatory	Type	Description
name	No	String	Component name.

Parameter	Mandatory	Type	Description
configurations	No	Array of Configuratio nItem objects	Component configuration item.

Table 4-19 ConfigurationItem

Parameter	Mandatory	Type	Description
name	No	String	Component configuration item name.
value	No	Object	Component configuration item value.

Table 4-20 ClusterStatus

Parameter	Mandatory	Type	Description
phase	No	String	<p>Cluster status. Options:</p> <ul style="list-style-type: none"> ● Available: The cluster is running properly. ● Unavailable: The cluster is exhibiting unexpected behavior. Manually delete it. ● ScalingUp: Nodes are being added to the cluster. ● ScalingDown: The cluster is being downsized to fewer nodes. ● Creating: The cluster is being created. ● Deleting: The cluster is being deleted. ● Upgrading: The cluster is being upgraded. ● Resizing: Cluster specifications are being changed. ● RollingBack: The cluster is being rolled back. ● RollbackFailed: The cluster rollback is abnormal. ● Hibernating: The cluster is being hibernated. ● Hibernation: The cluster is in hibernation. ● Awaking: The cluster is being woken up from hibernation. ● Empty: The cluster does not have any resources. This field is discarded. ● Error: Resources in the cluster are abnormal. Manually delete the cluster.

Parameter	Mandatory	Type	Description
jobID	No	String	<p>ID of the task associated with the cluster in the current state. Options:</p> <ul style="list-style-type: none"> • ID of the associated task returned when creating a cluster. You can use it to obtain the auxiliary tasks for creating a cluster. • ID of the associated task returned when a cluster fails to be deleted or is deleted. If this parameter is not empty, you can use the task ID to obtain the auxiliary tasks for deleting a cluster. <p>NOTE Tasks are short-lived. Do not use task information in scenarios such as cluster status determination.</p>
reason	No	String	Reason of cluster state change. This parameter is returned if the cluster is not in the Available state.
message	No	String	Detailed information about why the cluster changes to the current state. This parameter is returned if the cluster is not in the Available state.
endpoints	No	Array of ClusterEndpoints objects	Access address of kube-apiserver in the cluster.
isLocked	No	Boolean	The CBC resource is locked.
lockScene	No	String	Scenario where the CBC resource is locked.
lockSource	No	String	Resource locking.
lockSourceId	No	String	ID of the locked resource.
deleteOption	No	Object	Whether to delete configurations. This parameter is contained only in the response to the deletion request.

Parameter	Mandatory	Type	Description
deleteStatus	No	Object	Whether to delete the status information. This parameter is contained only in the response to the deletion request.

Table 4-21 ClusterEndpoints

Parameter	Mandatory	Type	Description
url	No	String	Access address of kube-apiserver in the cluster.
type	No	String	Type of the cluster access address. <ul style="list-style-type: none"> • Internal: address for internal network access • External: address for external network access

Response Parameters

Status code: 201

Table 4-22 Response body parameters

Parameter	Type	Description
kind	String	API type. The value is fixed at Cluster or cluster and cannot be changed.
apiVersion	String	API version. The value is fixed at v3 and cannot be changed.
metadata	ClusterMetadata object	Basic information about a cluster. Metadata is a collection of attributes.
spec	ClusterSpec object	Detailed description of the cluster. CCE creates or updates objects by defining or updating spec .
status	ClusterStatus object	Cluster status and job ID of the cluster creation job.

Table 4-23 ClusterMetadata

Parameter	Type	Description
name	String	Cluster name. Enter 4 to 128 characters, starting with a lowercase letter and not ending with a hyphen (-). Only lowercase letters, digits, and hyphens (-) are allowed.
uid	String	Cluster ID, which uniquely identifies a cluster. This ID is automatically generated after a cluster is created. Only the automatically generated ID will take effect. When you create a yearly/monthly cluster, no cluster ID will be returned in the response body.
alias	String	Alias of a cluster name displayed on the CCE console, and the name can be changed. Enter 4 to 128 characters, starting with a lowercase letter and not ending with a hyphen (-). Only lowercase letters, digits, and hyphens (-) are allowed. A cluster alias must be unique. In the request body for creating or updating a cluster, if the cluster alias is not specified or set to null, the cluster name will be used as the cluster alias. In the response body for obtaining a cluster, the cluster alias is returned. If it is not configured, the cluster name will be returned.
annotations	Map<String,String>	Cluster annotations, in the format of key-value pairs. <pre>"annotations": { "key1" : "value1", "key2" : "value2" }</pre> <p>NOTE</p> <ul style="list-style-type: none"> • annotations: Does not label or select objects. The metadata in annotations may be small or large, structured or unstructured, and may include characters that are not allowed in labels. • This field is not stored in the database and is used only to specify the add-ons to be installed in the cluster. • Install ICAgent during cluster creation by adding the key-value pair "cluster.install.addons.external/install":{"addonTemplateName":"icagent"}".
labels	Map<String,String>	Cluster labels, in the format of key-value pairs. NOTE The value of this field is automatically generated by the system and is used by the frontend to identify the features supported by the cluster during the upgrade. Customized values are invalid.

Parameter	Type	Description
creationTimes tamp	String	Time when the cluster was created.
updateTimest amp	String	Time when the cluster was updated.

Table 4-24 ClusterSpec

Parameter	Type	Description
category	String	Cluster type. Options: <ul style="list-style-type: none"> • CCE: CCE cluster CCE cluster supports hybrid deployment of VMs and BMSs, and heterogeneous nodes such as GPU and NPU nodes, allowing you to run your containers in a secure and stable container runtime environment based on a high-performance network model. • Turbo: CCE Turbo cluster One-stop, cost-effective CCE Turbo clusters run on the cloud native 2.0 infrastructure featuring hardware-software synergy for lossless networking, high security and reliability, and intelligent scheduling.
type	String	Master node architecture: <ul style="list-style-type: none"> • VirtualMachine: x86 • ARM64: Arm-based Kunpeng

Parameter	Type	Description
flavor	String	<p>Cluster specifications. Specifications of clusters 1.15 or later versions can be changed after they are created. For details, see Modifying Cluster Specifications. Options:</p> <ul style="list-style-type: none"> ● cce.s1.small: a small-scale CCE cluster with one master node and a maximum of 50 worker nodes ● cce.s1.medium: a medium-scale CCE cluster with one master node and a maximum of 200 worker nodes ● cce.s2.small: a small-scale CCE cluster with three master nodes and a maximum of 50 worker nodes ● cce.s2.medium: a medium-scale CCE cluster with three master nodes and a maximum of 200 worker nodes ● cce.s2.large: a large-scale CCE cluster with three master nodes and a maximum of 1,000 worker nodes ● cce.s2.xlarge: an ultra-large-scale CCE cluster with three master nodes and a maximum of 2,000 worker nodes <p>NOTE The fields in the parameters are described as follows:</p> <ul style="list-style-type: none"> ● s1: specifies a cluster with one master node. If the master node is faulty, the cluster will become unavailable, but running workloads in the cluster are not affected. ● s2: specifies an HA cluster with three master nodes. If one of the master nodes is faulty, the cluster is still available. ● dec: specifies a DeC CCE cluster. For example, cce.dec.s1.small specifies a small-scale, DeC CCE cluster with one master node and a maximum of 50 worker nodes. ● small: specifies that a cluster can manage a maximum of 50 worker nodes. ● medium: specifies that a cluster can manage a maximum of 200 worker nodes. ● large: specifies that a cluster can manage a maximum of 1,000 worker nodes. ● xlarge: specifies that a cluster can manage a maximum of 2,000 worker nodes.

Parameter	Type	Description
version	String	<p>Cluster version, which mirrors the baseline version of the Kubernetes community. The latest version is recommended.</p> <p>You can create clusters of two latest versions on the CCE console. To learn which cluster versions are available, log in to the CCE console, create a cluster, and check the Cluster Version parameter.</p> <p>You can call APIs to create clusters of other versions. However, these cluster versions will be gradually terminated. For details about the support policy, see the CCE announcement.</p> <p>NOTE</p> <ul style="list-style-type: none"> • If not specified, a cluster of the latest version will be created. • If a baseline cluster version is specified but the R version is not specified, a cluster of the latest R version will be created by default. It is a good practice not to specify the R version. • CCE Turbo clusters of v1.19 or later are commercially available.
platformVersion	String	<p>CCE cluster platform version, indicating the internal version under the cluster version (version). Platform versions are used to trace iterations in a major cluster version. They are unique within a major cluster version and recounted when the major cluster version changes. This parameter cannot be customized. When you create a cluster, the latest corresponding platform version is automatically selected.</p> <p>The format of platformVersion is cce.X.Y.</p> <ul style="list-style-type: none"> • X: internal feature version, indicating changes in features, patches, or OS support in the cluster version. The value starts from 1 and increases monotonically. • Y: patch version of an internal feature version. It is used only for software package update after the feature version is brought online. No other modification is involved. The value starts from 0 and increases monotonically.

Parameter	Type	Description
description	String	Cluster description, for example, which purpose the cluster is intended to serve. By default, this field is left unspecified. To modify cluster description after the cluster is created, call the API for updating cluster information or go to the cluster details page on the CCE console. Only UTF-8 encoding is supported.
customSan	Array of strings	Custom SAN field in the server certificate of the cluster API server, which must comply with the SSL and X509 format specifications. <ol style="list-style-type: none"> 1. Duplicate names are not allowed. 2. Must comply with the IP address and domain name formats. Example: SAN 1: DNS Name=example.com SAN 2: DNS Name=www.example.com SAN 3: DNS Name=example.net SAN 4: IP Address=93.184.216.34
ipv6enable	Boolean	Whether the cluster supports IPv6 addresses. This field is supported in clusters of v1.15 and later versions.
hostNetwork	HostNetwork object	Node networking parameters, including VPC and subnet ID. This field is mandatory because nodes in a cluster communicate with each other by using a VPC.
containerNetwork	ContainerNetwork object	Container networking parameters, including the container network model and container CIDR block.
eniNetwork	EniNetwork object	Configuration of the Cloud Native Network 2.0 model. Specify this field when creating a CCE Turbo cluster.
serviceNetwork	ServiceNetwork object	Service CIDR block, including IPv4 CIDR blocks.
authentication	Authentication object	Configurations of the cluster authentication mode.
billingMode	Integer	Billing mode of a cluster. <ul style="list-style-type: none"> • 0: pay-per-use • 1: yearly/monthly Defaults to pay-per-use.
masters	Array of MasterSpec objects	Advanced configurations of master nodes

Parameter	Type	Description
kubernetesSvcIpRange	String	Service CIDR blocks for Kubernetes cluster IPs. This field is available only for clusters of v1.11.7 and later. If this parameter is not specified during cluster creation, the default value 10.247.0.0/16 will be used. This parameter is deprecated. Use serviceNetwork instead. The new field contains the IPv4 CIDR blocks.
clusterTags	Array of ResourceTag objects	Cluster resource tags.
kubeProxyMode	String	Service forwarding mode. Options: <ul style="list-style-type: none"> • iptables: Traditional kube-proxy uses iptables rules to implement Service load balancing. In this mode, too many iptables rules will be generated when many Services are deployed. In addition, non-incremental updates will cause latency and even tangible performance issues in the case of service traffic spikes. • ipvs: Optimized kube-proxy mode with higher throughput and faster speed. This mode supports incremental updates and can keep connections uninterrupted during Service updates. It is suitable for large-sized clusters. <p>NOTE iptables is used by default.</p>
az	String	AZ. This field is returned only for a query. For details about AZs supported by CCE, see Regions and Endpoints .
extendParam	ClusterExtendedParam object	Extended field to decide whether the cluster will span across AZs or belong to a specified enterprise project, or whether a dedicated CCE cluster is to be created.
supportIstio	Boolean	Whether Istio is supported.
configurationsOverride	Array of PackageConfiguration objects	Cluster default component configuration override. If you specify a component or parameter that is not supported, the configuration item will be ignored. For details about the supported components and their parameters, see Cluster Configuration Management .

Table 4-25 HostNetwork

Parameter	Type	Description
vpc	String	<p>ID of the VPC used to create a master node. You can obtain it in either of the following ways:</p> <ul style="list-style-type: none"> • Method 1: Log in to the VPC console and view the VPC ID in the VPC details page. • Method 2: Use the VPC API to obtain VPC IDs. <p>For details, see Querying VPCs.</p>
subnet	String	<p>Network ID of the subnet used to create a master node. You can obtain it in either of the following ways:</p> <ul style="list-style-type: none"> • Method 1: Log in to the VPC console and click the target subnet on the Subnets page. You can view the network ID on the displayed page. • Method 2: Use the VPC API to obtain subnets. <p>For details, see Querying Subnets.</p>
SecurityGroup	String	<p>Default worker node security group ID of the cluster. If specified, the cluster will be bound to the target security group. Otherwise, the system will automatically create a default worker node security group for you. The default worker node security group needs to allow access from certain ports to ensure normal communications. For details, see How Do I Harden the Automatically Created Security Group Rules for CCE Cluster Nodes.</p>

Table 4-26 ContainerNetwork

Parameter	Type	Description
mode	String	<p>Container network model. Select one of the following possible values:</p> <ul style="list-style-type: none"> • overlay_l2: an overlay_l2 network (container tunnel network) built for containers by using OpenVSwitch (OVS). • vpc-router: an underlay_l2 network built for containers by using IPvlan and custom VPC routes. • eni: Cloud Native Network 2.0. This model has integrated cloud native elastic network interfaces (ENIs), uses VPC CIDR blocks to allocate container IP addresses, and allows direct traffic distribution to containers through a load balancer for high performance. Use this model when creating a CCE Turbo cluster.
cidr	String	<p>Container CIDR block. Recommended: 10.0.0.0/12-19, 172.16.0.0/16-19, or 192.168.0.0/16-19. If the selected CIDR block conflicts with existing ones, an error will be reported.</p> <p>Not editable after the cluster is created. (deprecated. A specified cidrs will make cidr invalid.)</p>
cidrs	Array of ContainerCIDR objects	<p>List of container CIDR blocks. In clusters of v1.21 or later, the cidrs field is used. When the cluster network type is vpc-router, you can configure a maximum of 20 container CIDR blocks. In cluster versions earlier than v1.21, if the cidrs field is used, the first CIDR element in the array is used as the container CIDR block.</p> <p>The configuration cannot be modified after the cluster is created.</p>

Table 4-27 ContainerCIDR

Parameter	Type	Description
cidr	String	<p>Container CIDR block. Recommended: 10.0.0.0/12-19, 172.16.0.0/16-19, and 192.168.0.0/16-19</p>

Table 4-28 EniNetwork

Parameter	Type	Description
eniSubnetId	String	IPv4 subnet ID of ENI subnet. (IPv6 is not supported and is being discarded.) You can obtain it in either of the following ways: <ul style="list-style-type: none"> Method 1: Log in to the VPC console and click the target subnet on the Subnets page. You can view the IPv4 subnet ID on the displayed page. Method 2: Use the VPC API to obtain subnet IDs. For details, see Querying Subnets .
eniSubnetCIDR	String	ENI subnet CIDR (being discarded)
subnets	Array of NetworkSubnet objects	List of IPv4 subnet IDs

Table 4-29 NetworkSubnet

Parameter	Type	Description
subnetID	String	IPv4 subnet ID of the subnet for creating master nodes. IPv6 is not supported. You can obtain it in either of the following ways: <ul style="list-style-type: none"> Method 1: Log in to the VPC console and click the target subnet on the Subnets page. You can view the IPv4 subnet ID on the displayed page. Method 2: Use the VPC API to obtain subnets. For details, see Querying Subnets .

Table 4-30 ServiceNetwork

Parameter	Type	Description
IPv4CIDR	String	Value range of the Kubernetes clusterIP IPv4 CIDR blocks. If this parameter is not specified during cluster creation, the default value 10.247.0.0/16 will be used.

Table 4-31 Authentication

Parameter	Type	Description
mode	String	Cluster authentication mode. <ul style="list-style-type: none"> Clusters of Kubernetes v1.11 or earlier support x509, rbac, and authenticating_proxy. Defaults to x509. Clusters of Kubernetes v1.13 or later support rbac and authenticating_proxy. Defaults to rbac.
authenticatingProxy	AuthenticatingProxy object	Configuration related to the <code>authenticating_proxy</code> mode. This field is mandatory when the authentication mode is authenticating_proxy .

Table 4-32 AuthenticatingProxy

Parameter	Type	Description
ca	String	X509 CA certificate (Base64-encoded) configured in <code>authenticating_proxy</code> mode. This field is mandatory when the cluster authentication mode is authenticating_proxy . Maximum size: 1 MB
cert	String	Client certificate issued by the X509 CA certificate configured in <code>authenticating_proxy</code> mode, which is used for authentication from kube-apiserver to the extended API server. (The value must be Base64-encoded.) This field is mandatory when the cluster authentication mode is authenticating_proxy .
privateKey	String	Private key of the client certificate issued by the X509 CA certificate configured in <code>authenticating_proxy</code> mode, which is used for authentication from kube-apiserver to the extended API server. The private key used by the Kubernetes cluster does not support password encryption. Use an unencrypted private key. (The value must be Base64-encoded.) This field is mandatory when the cluster authentication mode is authenticating_proxy .

Table 4-33 MasterSpec

Parameter	Type	Description
availabilityZone	String	AZ

Table 4-34 ResourceTag

Parameter	Type	Description
key	String	Key. <ul style="list-style-type: none"> • Cannot be null. Max characters: 128. • Use letters, digits, and spaces in UTF-8 format. • Can contain the following special characters: <code>._:/=-@</code>. • Cannot start with <code>_sys_</code>.
value	String	Value. <ul style="list-style-type: none"> • Can be null but not the default. Max characters: 255. • Use letters, digits, and spaces in UTF-8 format. • Can contain the following special characters: <code>._:/=-@</code>.

Table 4-35 ClusterExtendParam

Parameter	Type	Description
clusterAZ	String	AZ of master nodes in a cluster. For details about AZs supported by CCE, see Regions and Endpoints . <ul style="list-style-type: none"> • multi_az: (Optional) The cluster will span across AZs. Multiple AZs can be configured only when a cluster with multiple master nodes is used. • <i>AZ of the dedicated cloud computing pool</i>: The cluster will be deployed in the DeC AZ. It is mandatory for DeC CCE clusters.

Parameter	Type	Description
dssMasterVolumes	String	<p>Whether the system and data disks of a master node use dedicated distributed storage. If this parameter is omitted or left unspecified, EVS disks are used by default.</p> <p>This parameter is mandatory for dedicated CCE clusters. It is in the following format: <code><rootVol.dssPoolID>.<rootVol.volType>;<dataVol.dssPoolID>.<dataVol.volType></code></p> <p>Field description:</p> <ul style="list-style-type: none"> • rootVol is the system disk. dataVol is the data disk. • dssPoolID indicates the ID of the DSS storage pool. • volType indicates the storage volume type of the DSS storage pool, such as SAS and SSD. <p>Example: <code>c950ee97-587c-4f24-8a74-3367e3da570f.sas;6edbc2f4-1507-44f8-ac0d-eed1d2608d38.ssd</code></p> <p>NOTE This field cannot be configured for non-dedicated CCE clusters.</p>
enterpriseProjectId	String	<p>ID of the enterprise project that a cluster belongs to.</p> <p>NOTE</p> <ul style="list-style-type: none"> • An enterprise project can be configured only after the enterprise project function is enabled. • The enterprise project to which the cluster belongs must be the same as that to which other cloud service resources associated with the cluster belong.

Parameter	Type	Description
kubeProxyMode	String	<p>Service forwarding mode. Two modes are available:</p> <ul style="list-style-type: none"> • iptables: Traditional kube-proxy uses iptables rules to implement Service load balancing. In this mode, too many iptables rules will be generated when many Services are deployed. In addition, non-incremental updates will cause latency and even tangible performance issues in the case of service traffic spikes. • ipvs: Optimized kube-proxy mode with higher throughput and faster speed. This mode supports incremental updates and can keep connections uninterrupted during Service updates. It is suitable for large-sized clusters. <p>NOTE This parameter has been deprecated. If this parameter and kubeProxyMode in ClusterSpec are specified at the same time, the latter is used.</p>
clusterExternalIP	String	EIP of the master node
alpha.cce/fixPoolMask	String	<p>Number of mask bits of the fixed IP address pool of the container network model. This field is supported only for the VPC network model (vpc-router).</p> <p>This parameter determines the number of container IP addresses that can be allocated to a node. The maximum number of pods that can be created on a node is decided by this parameter and maxPods set during node creation.</p> <p>For details, see Maximum Number of Pods That Can Be Created on a Node.</p> <p>For integer characters, the value ranges from 24 to 28.</p>
decMasterFlavor	String	Specifications of the master node in the dedicated hybrid cluster.
dockerUmaskMode	String	Default UmaskMode configuration of Docker in a cluster. The value can be secure or normal . If this parameter is not specified, normal is used by default.

Parameter	Type	Description
kubernetes.io/cpuManagerPolicy	String	<p>Cluster CPU management policy. The value can be none (or null) or static. The default value is none (or null).</p> <ul style="list-style-type: none"> ● none or null: disables pods from exclusively occupying CPUs. Select this option if you want a large pool of shareable CPU cores. ● static: enables pods to exclusively occupy CPUs. Select this option if your workload is sensitive to CPU cache and scheduling latency. In a CCE Turbo cluster, this setting is valid only for nodes where common containers, not Kata containers, run.
orderId	String	<p>Order ID. This parameter is returned in the response when the cluster is billed on a yearly/monthly basis with auto payment enabled (only in creation scenarios).</p>
periodType	String	<ul style="list-style-type: none"> ● month: The unit is month. ● year: The unit is year. <p>NOTE Request parameter, which is valid and mandatory when billingMode is set to 1 (yearly/monthly billing). Response parameter, which is returned only when a yearly/monthly cluster is created.</p>
periodNum	Integer	<p>Subscription duration. The value can be:</p> <ul style="list-style-type: none"> ● If periodType is month, the value ranges from 1 to 9. ● If periodType is year, the value ranges from 1 to 3. <p>NOTE Request parameter, which is valid and mandatory when billingMode is set to 1. Response parameter, which is returned only when a yearly/monthly cluster is created.</p>
isAutoRenew	String	<p>Whether auto renewal is enabled.</p> <ul style="list-style-type: none"> ● true: Auto renewal is enabled. ● false: Auto renewal is not enabled. <p>NOTE This field is valid when billingMode is set to 1. If not specified, auto renewal is not enabled.</p>

Parameter	Type	Description
isAutoPay	String	Whether to deduct fees automatically. <ul style="list-style-type: none"> • true: Enable automatic fee deduction. • false: Do not enable automatic fee deduction. NOTE This field is valid when billingMode is set to 1 . If not specified, auto fee deduction is not enabled.
upgradefrom	String	Records of how the cluster is upgraded to the current version

Table 4-36 PackageConfiguration

Parameter	Type	Description
name	String	Component name.
configurations	Array of ConfigurationItem objects	Component configuration item.

Table 4-37 ConfigurationItem

Parameter	Type	Description
name	String	Component configuration item name.
value	Object	Component configuration item value.

Table 4-38 ClusterStatus

Parameter	Type	Description
phase	String	<p>Cluster status. Options:</p> <ul style="list-style-type: none"> ● Available: The cluster is running properly. ● Unavailable: The cluster is exhibiting unexpected behavior. Manually delete it. ● ScalingUp: Nodes are being added to the cluster. ● ScalingDown: The cluster is being downsized to fewer nodes. ● Creating: The cluster is being created. ● Deleting: The cluster is being deleted. ● Upgrading: The cluster is being upgraded. ● Resizing: Cluster specifications are being changed. ● RollingBack: The cluster is being rolled back. ● RollbackFailed: The cluster rollback is abnormal. ● Hibernating: The cluster is being hibernated. ● Hibernation: The cluster is in hibernation. ● Awaking: The cluster is being woken up from hibernation. ● Empty: The cluster does not have any resources. This field is discarded. ● Error: Resources in the cluster are abnormal. Manually delete the cluster.
jobID	String	<p>ID of the task associated with the cluster in the current state. Options:</p> <ul style="list-style-type: none"> ● ID of the associated task returned when creating a cluster. You can use it to obtain the auxiliary tasks for creating a cluster. ● ID of the associated task returned when a cluster fails to be deleted or is deleted. If this parameter is not empty, you can use the task ID to obtain the auxiliary tasks for deleting a cluster. <p>NOTE Tasks are short-lived. Do not use task information in scenarios such as cluster status determination.</p>
reason	String	Reason of cluster state change. This parameter is returned if the cluster is not in the Available state.

Parameter	Type	Description
message	String	Detailed information about why the cluster changes to the current state. This parameter is returned if the cluster is not in the Available state.
endpoints	Array of ClusterEndpoints objects	Access address of kube-apiserver in the cluster.
isLocked	Boolean	The CBC resource is locked.
lockScene	String	Scenario where the CBC resource is locked.
lockSource	String	Resource locking.
lockSourceId	String	ID of the locked resource.
deleteOption	Object	Whether to delete configurations. This parameter is contained only in the response to the deletion request.
deleteStatus	Object	Whether to delete the status information. This parameter is contained only in the response to the deletion request.

Table 4-39 ClusterEndpoints

Parameter	Type	Description
url	String	Access address of kube-apiserver in the cluster.
type	String	Type of the cluster access address. <ul style="list-style-type: none"> • Internal: address for internal network access • External: address for external network access

Example Requests

- Create an HA CCE cluster of v1.19 with three master nodes and a maximum of 50 worker nodes billed on a pay-per-use basis.

```
/api/v3/projects/{project_id}/clusters
```

```
{
  "kind": "Cluster",
  "apiVersion": "v3",
  "metadata": {
    "name": "cluster"
  },
  "spec": {
    "category": "CCE",
    "flavor": "cce.s2.small",
```

```

"version": "v1.19",
"hostNetwork": {
  "vpc": "030bfb19-5fa7-42ad-8a0d-c0721d268867",
  "subnet": "ca964acf-8468-4735-8229-97940ef6c881"
},
"containerNetwork": {
  "mode": "vpc-router",
  "cidr": "10.0.0.0/16"
},
"kubernetesSvcIppRange": "10.247.0.0/16",
"description": "",
"billingMode": 0,
"extendParam": {
  "kubeProxyMode": "iptables",
  "alpha.cce/fixPoolMask": "25",
  "enterpriseProjectId": "0"
},
"authentication": {
  "mode": "rbac"
},
"ipv6enable": false
}

```

- Create an HA CCE cluster of v1.19 with three master nodes and a maximum of 50 worker nodes billed on a pay-per-use basis and install ICAGENT in the cluster.

/api/v3/projects/{project_id}/clusters

```

{
  "kind": "Cluster",
  "apiVersion": "v3",
  "metadata": {
    "name": "cluster",
    "annotations": {
      "cluster.install.addons.external/install": "[{\\"addonTemplateName\\":\\"icagent\\"}]"
    }
  },
  "spec": {
    "category": "CCE",
    "flavor": "cce.s2.small",
    "version": "v1.19",
    "hostNetwork": {
      "vpc": "030bfb19-5fa7-42ad-8a0d-c0721d268867",
      "subnet": "ca964acf-8468-4735-8229-97940ef6c881"
    },
    "containerNetwork": {
      "mode": "vpc-router",
      "cidr": "10.0.0.0/16"
    },
    "kubernetesSvcIppRange": "10.247.0.0/16",
    "description": "",
    "billingMode": 0,
    "extendParam": {
      "kubeProxyMode": "iptables",
      "alpha.cce/fixPoolMask": "25",
      "enterpriseProjectId": "0"
    },
    "authentication": {
      "mode": "rbac"
    },
    "ipv6enable": false
  }
}

```

- Create an HA CCE cluster of v1.19 with three master nodes and a maximum of 50 worker nodes billed on a pay-per-use basis and configure the default security group for custom nodes in the cluster.

```
/api/v3/projects/{project_id}/clusters
{
  "kind": "Cluster",
  "apiVersion": "v3",
  "metadata": {
    "name": "cluster"
  },
  "spec": {
    "category": "CCE",
    "flavor": "cce.s2.small",
    "version": "v1.19",
    "hostNetwork": {
      "vpc": "030bfb19-5fa7-42ad-8a0d-c0721d268867",
      "subnet": "ca964acf-8468-4735-8229-97940ef6c881",
      "SecurityGroup": "a4ef108c-2ec6-492f-a6c4-7b64e25ae490"
    },
    "containerNetwork": {
      "mode": "vpc-router",
      "cidr": "10.0.0.0/16"
    },
    "kubernetesSvcIpRange": "10.247.0.0/16",
    "description": "",
    "billingMode": 0,
    "extendParam": {
      "kubeProxyMode": "iptables",
      "alpha.cce/fixPoolMask": "25",
      "enterpriseProjectId": "0"
    },
    "authentication": {
      "mode": "rbac"
    },
    "ipv6enable": false
  }
}
```

- Create an HA CCE cluster of v1.25 with three master nodes and a maximum of 50 worker nodes billed on a pay-per-use basis.

```
/api/v3/projects/{project_id}/clusters
{
  "kind": "Cluster",
  "apiVersion": "v3",
  "metadata": {
    "name": "cluster"
  },
  "spec": {
    "category": "Turbo",
    "flavor": "cce.s2.small",
    "version": "v1.25",
    "type": "VirtualMachine",
    "hostNetwork": {
      "vpc": "030bfb19-5fa7-42ad-8a0d-c0721d268867",
      "subnet": "ca964acf-8468-4735-8229-97940ef6c881"
    },
    "containerNetwork": {
      "mode": "eni"
    },
    "eniNetwork": {
      "eniSubnetId": "861fb11d-2f0e-4c10-a98a-166dc26e4ff7",
      "eniSubnetCIDR": "192.168.0.0/24",
      "subnets": [ {
        "subnetID": "861fb11d-2f0e-4c10-a98a-166dc26e4ff7"
      } ]
    },
    "serviceNetwork": {
      "IPv4CIDR": "10.247.0.0/16"
    },
    "description": ""
  }
}
```

```
"billingMode" : 0,
"extendParam" : {
  "kubeProxyMode" : "iptables",
  "enterpriseProjectId" : "0"
},
"authentication" : {
  "mode" : "rbac"
},
"ipv6enable" : false
}
```

Example Responses

Status code: 201

The cluster creation job is successfully delivered.

```
{
  "kind" : "Cluster",
  "apiVersion" : "v3",
  "metadata" : {
    "alias" : "cluster",
    "name" : "cluster",
    "uid" : "bce956e1-87f3-11ec-b5e5-0255ac101514",
    "creationTimestamp" : "2022-02-07 08:55:45.785425492 +0000 UTC",
    "updateTimestamp" : "2022-02-07 08:55:45.78542824 +0000 UTC",
    "annotations" : {
      "jobid" : "bd42f724-87f3-11ec-b5e5-0255ac101514",
      "resourceJobId" : "bce99f78-87f3-11ec-b5e5-0255ac101514"
    }
  },
  "spec" : {
    "category" : "CCE",
    "type" : "VirtualMachine",
    "flavor" : "cce.s2.small",
    "version" : "v1.19.10-r0",
    "platformVersion" : "cce.5.0",
    "hostNetwork" : {
      "vpc" : "030bfb19-5fa7-42ad-8a0d-c0721d268867",
      "subnet" : "ca964acf-8468-4735-8229-97940ef6c881"
    },
    "containerNetwork" : {
      "mode" : "vpc-router",
      "cidr" : "10.0.0/16",
      "cidrs" : [ {
        "cidr" : "10.0.0/16"
      } ]
    },
    "eniNetwork" : { },
    "authentication" : {
      "mode" : "rbac",
      "authenticatingProxy" : { }
    },
    "billingMode" : 0,
    "kubernetesSvcIpsRange" : "10.247.0.0/16",
    "kubeProxyMode" : "iptables",
    "extendParam" : {
      "alpha.cce/fixPoolMask" : "25",
      "enterpriseProjectId" : "0",
      "kubeProxyMode" : "iptables",
      "orderID" : ""
    }
  },
  "status" : {
    "phase" : "Creating",
    "jobID" : "bd42f724-87f3-11ec-b5e5-0255ac101514"
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

- Create an HA CCE cluster of v1.19 with three master nodes and a maximum of 50 worker nodes billed on a pay-per-use basis.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class CreateClusterSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateClusterRequest request = new CreateClusterRequest();
        Cluster body = new Cluster();
        ClusterExtendParam extendParamSpec = new ClusterExtendParam();
        extendParamSpec.withEnterpriseProjectId("0")
            .withKubeProxyMode("iptables")
            .withAlphaCceFixPoolMask("25");
        Authentication authenticationSpec = new Authentication();
        authenticationSpec.withMode("rbac");
        ContainerNetwork containerNetworkSpec = new ContainerNetwork();
        containerNetworkSpec.withMode(ContainerNetwork.ModeEnum.fromValue("vpc-router"))
            .withCidr("10.0.0.0/16");
        HostNetwork hostNetworkSpec = new HostNetwork();
        hostNetworkSpec.withVpc("030bfb19-5fa7-42ad-8a0d-c0721d268867")
            .withSubnet("ca964acf-8468-4735-8229-97940ef6c881");
        ClusterSpec specbody = new ClusterSpec();
        specbody.withCategory(ClusterSpec.CategoryEnum.fromValue("CCE"))
            .withFlavor("cce.s2.small")
            .withVersion("v1.19")
            .withDescription("")
            .withIpv6enable(false)
            .withHostNetwork(hostNetworkSpec)
            .withContainerNetwork(containerNetworkSpec)
            .withAuthentication(authenticationSpec)
            .withBillingMode(0)
            .withKubernetesSvclpRange("10.247.0.0/16")
            .withExtendParam(extendParamSpec);
```

```
ClusterMetadata metadatabody = new ClusterMetadata();
metadatabody.setName("cluster");
body.withSpec(specbody);
body.withMetadata(metadatabody);
body.withApiVersion("v3");
body.withKind("Cluster");
request.withBody(body);
try {
    CreateClusterResponse response = client.createCluster(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

- Create an HA CCE cluster of v1.19 with three master nodes and a maximum of 50 worker nodes billed on a pay-per-use basis and install ICAgent in the cluster.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

import java.util.Map;
import java.util.HashMap;

public class CreateClusterSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();

        CreateClusterRequest request = new CreateClusterRequest();
        Cluster body = new Cluster();
        ClusterExtendParam extendParamSpec = new ClusterExtendParam();
        extendParamSpec.withEnterpriseProjectId("0")
            .withKubeProxyMode("iptables")
            .withAlphaCceFixPoolMask("25");
```

```

Authentication authenticationSpec = new Authentication();
authenticationSpec.withMode("rbac");
ContainerNetwork containerNetworkSpec = new ContainerNetwork();
containerNetworkSpec.withMode(ContainerNetwork.ModeEnum.fromValue("vpc-router"))
    .withCidr("10.0.0.0/16");
HostNetwork hostNetworkSpec = new HostNetwork();
hostNetworkSpec.withVpc("030bfb19-5fa7-42ad-8a0d-c0721d268867")
    .withSubnet("ca964acf-8468-4735-8229-97940ef6c881");
ClusterSpec specbody = new ClusterSpec();
specbody.withCategory(ClusterSpec.CategoryEnum.fromValue("CCE"))
    .withFlavor("cce.s2.small")
    .withVersion("v1.19")
    .withDescription("")
    .withIpv6enable(false)
    .withHostNetwork(hostNetworkSpec)
    .withContainerNetwork(containerNetworkSpec)
    .withAuthentication(authenticationSpec)
    .withBillingMode(0)
    .withKubernetesSvclpRange("10.247.0.0/16")
    .withExtendParam(extendParamSpec);
Map<String, String> listMetadataAnnotations = new HashMap<>();
listMetadataAnnotations.put("cluster.install.addons.external/install",
"["addonTemplateName":"icagent"]");
ClusterMetadata metadatabody = new ClusterMetadata();
metadatabody.withName("cluster")
    .withAnnotations(listMetadataAnnotations);
body.withSpec(specbody);
body.withMetadata(metadatabody);
body.withApiVersion("v3");
body.withKind("Cluster");
request.withBody(body);
try {
    CreateClusterResponse response = client.createCluster(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
}

```

- Create an HA CCE cluster of v1.19 with three master nodes and a maximum of 50 worker nodes billed on a pay-per-use basis and configure the default security group for custom nodes in the cluster.

```

package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class CreateClusterSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
    }
}

```



```
// In this example, AK and SK are stored in environment variables for authentication. Before
// running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
// environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

CceClient client = CceClient.newBuilder()
    .withCredential(auth)
    .withRegion(CceRegion.valueOf("<YOUR REGION>"))
    .build();

CreateClusterRequest request = new CreateClusterRequest();
Cluster body = new Cluster();
ClusterExtendParam extendParamSpec = new ClusterExtendParam();
extendParamSpec.withEnterpriseProjectId("0")
    .withKubeProxyMode("iptables")
    .withAlphaCceFixPoolMask("25");
Authentication authenticationSpec = new Authentication();
authenticationSpec.withMode("rbac");
ContainerNetwork containerNetworkSpec = new ContainerNetwork();
containerNetworkSpec.withMode(ContainerNetwork.ModeEnum.fromValue("vpc-router"))
    .withCidr("10.0.0.0/16");
HostNetwork hostNetworkSpec = new HostNetwork();
hostNetworkSpec.withVpc("030bfb19-5fa7-42ad-8a0d-c0721d268867")
    .withSubnet("ca964acf-8468-4735-8229-97940ef6c881")
    .withSecurityGroup("a4ef108c-2ec6-492f-a6c4-7b64e25ae490");
ClusterSpec specbody = new ClusterSpec();
specbody.withCategory(ClusterSpec.CategoryEnum.fromValue("CCE"))
    .withFlavor("cce.s2.small")
    .withVersion("v1.19")
    .withDescription("")
    .withIpv6enable(false)
    .withHostNetwork(hostNetworkSpec)
    .withContainerNetwork(containerNetworkSpec)
    .withAuthentication(authenticationSpec)
    .withBillingMode(0)
    .withKubernetesSvclpRange("10.247.0.0/16")
    .withExtendParam(extendParamSpec);
ClusterMetadata metadatabody = new ClusterMetadata();
metadatabody.withName("cluster");
body.withSpec(specbody);
body.withMetadata(metadatabody);
body.withApiVersion("v3");
body.withKind("Cluster");
request.withBody(body);
try {
    CreateClusterResponse response = client.createCluster(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

- Create an HA CCE cluster of v1.25 with three master nodes and a maximum of 50 worker nodes billed on a pay-per-use basis.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateClusterSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();

        CreateClusterRequest request = new CreateClusterRequest();
        Cluster body = new Cluster();
        ClusterExtendParam extendParamSpec = new ClusterExtendParam();
        extendParamSpec.withEnterpriseProjectId("0")
            .withKubeProxyMode("iptables");
        Authentication authenticationSpec = new Authentication();
        authenticationSpec.withMode("rbac");
        ServiceNetwork serviceNetworkSpec = new ServiceNetwork();
        serviceNetworkSpec.withIpv4CIDR("10.247.0.0/16");
        List<NetworkSubnet> listEniNetworkSubnets = new ArrayList<>();
        listEniNetworkSubnets.add(
            new NetworkSubnet()
                .withSubnetID("861fb11d-2f0e-4c10-a98a-166dc26e4ff7")
        );
        EniNetwork eniNetworkSpec = new EniNetwork();
        eniNetworkSpec.withEniSubnetID("861fb11d-2f0e-4c10-a98a-166dc26e4ff7")
            .withEniSubnetCIDR("192.168.0.0/24")
            .withSubnets(listEniNetworkSubnets);
        ContainerNetwork containerNetworkSpec = new ContainerNetwork();
        containerNetworkSpec.withMode(ContainerNetwork.ModeEnum.fromValue("eni"));
        HostNetwork hostNetworkSpec = new HostNetwork();
        hostNetworkSpec.withVpc("030bfb19-5fa7-42ad-8a0d-c0721d268867")
            .withSubnet("ca964acf-8468-4735-8229-97940ef6c881");
        ClusterSpec specbody = new ClusterSpec();
        specbody.withCategory(ClusterSpec.CategoryEnum.fromValue("Turbo"))
            .withType(ClusterSpec.TypeEnum.fromValue("VirtualMachine"))
            .withFlavor("cce.s2.small")
            .withVersion("v1.25")
            .withDescription("")
            .withIpv6enable(false)
            .withHostNetwork(hostNetworkSpec)
            .withContainerNetwork(containerNetworkSpec)
            .withEniNetwork(eniNetworkSpec)
    }
}
```

```
.withServiceNetwork(serviceNetworkSpec)
.withAuthentication(authenticationSpec)
.withBillingMode(0)
.withExtendParam(extendParamSpec);
ClusterMetadata metadatabody = new ClusterMetadata();
metadatabody.withName("cluster");
body.withSpec(specbody);
body.withMetadata(metadatabody);
body.withApiVersion("v3");
body.withKind("Cluster");
request.withBody(body);
try {
    CreateClusterResponse response = client.createCluster(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

- Create an HA CCE cluster of v1.19 with three master nodes and a maximum of 50 worker nodes billed on a pay-per-use basis.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateClusterRequest()
        extendParamSpec = ClusterExtendParam(
            enterprise_project_id="0",
            kube_proxy_mode="iptables",
            alpha_cce_fix_pool_mask="25"
        )
        authenticationSpec = Authentication(
            mode="rbac"
        )
        containerNetworkSpec = ContainerNetworkSpec(
```

```

        mode="vpc-router",
        cidr="10.0.0.0/16"
    )
    hostNetworkSpec = HostNetwork(
        vpc="030bfb19-5fa7-42ad-8a0d-c0721d268867",
        subnet="ca964acf-8468-4735-8229-97940ef6c881"
    )
    specbody = ClusterSpec(
        category="CCE",
        flavor="cce.s2.small",
        version="v1.19",
        description="",
        ipv6enable=False,
        host_network=hostNetworkSpec,
        container_network=containerNetworkSpec,
        authentication=authenticationSpec,
        billing_mode=0,
        kubernetes_svc_ip_range="10.247.0.0/16",
        extend_param=extendParamSpec
    )
    metadatabody = ClusterMetadata(
        name="cluster"
    )
    request.body = Cluster(
        spec=specbody,
        metadata=metadatabody,
        api_version="v3",
        kind="Cluster"
    )
    response = client.create_cluster(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)

```

- Create an HA CCE cluster of v1.19 with three master nodes and a maximum of 50 worker nodes billed on a pay-per-use basis and install ICAgent in the cluster.

```

# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateClusterRequest()
        extendParamSpec = ClusterExtendParam(
            enterprise_project_id="0",

```

```

        kube_proxy_mode="iptables",
        alpha_cce_fix_pool_mask="25"
    )
    authenticationSpec = Authentication(
        mode="rbac"
    )
    containerNetworkSpec = ContainerNetwork(
        mode="vpc-router",
        cidr="10.0.0.0/16"
    )
    hostNetworkSpec = HostNetwork(
        vpc="030bfb19-5fa7-42ad-8a0d-c0721d268867",
        subnet="ca964acf-8468-4735-8229-97940ef6c881"
    )
    specbody = ClusterSpec(
        category="CCE",
        flavor="cce.s2.small",
        version="v1.19",
        description="",
        ipv6enable=False,
        host_network=hostNetworkSpec,
        container_network=containerNetworkSpec,
        authentication=authenticationSpec,
        billing_mode=0,
        kubernetes_svc_ip_range="10.247.0.0/16",
        extend_param=extendParamSpec
    )
    listAnnotationsMetadata = {
        "cluster.install.addons.external/install": "[{"addonTemplateName": "icagent"}]"
    }
    metadatabody = ClusterMetadata(
        name="cluster",
        annotations=listAnnotationsMetadata
    )
    request.body = Cluster(
        spec=specbody,
        metadata=metadatabody,
        api_version="v3",
        kind="Cluster"
    )
    response = client.create_cluster(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)

```

- Create an HA CCE cluster of v1.19 with three master nodes and a maximum of 50 worker nodes billed on a pay-per-use basis and configure the default security group for custom nodes in the cluster.

```

# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

```

```

credentials = BasicCredentials(ak, sk, projectId) \

client = CceClient.new_builder() \
.with_credentials(credentials) \
.with_region(CceRegion.value_of("<YOUR REGION>")) \
.build()

try:
request = CreateClusterRequest()
extendParamSpec = ClusterExtendParam(
    enterprise_project_id="0",
    kube_proxy_mode="iptables",
    alpha_cce_fix_pool_mask="25"
)
authenticationSpec = Authentication(
    mode="rbac"
)
containerNetworkSpec = ContainerNetwork(
    mode="vpc-router",
    cidr="10.0.0.0/16"
)
hostNetworkSpec = HostNetwork(
    vpc="030bfb19-5fa7-42ad-8a0d-c0721d268867",
    subnet="ca964acf-8468-4735-8229-97940ef6c881",
    security_group="a4ef108c-2ec6-492f-a6c4-7b64e25ae490"
)
specbody = ClusterSpec(
    category="CCE",
    flavor="cce.s2.small",
    version="v1.19",
    description="",
    ipv6enable=False,
    host_network=hostNetworkSpec,
    container_network=containerNetworkSpec,
    authentication=authenticationSpec,
    billing_mode=0,
    kubernetes_svc_ip_range="10.247.0.0/16",
    extend_param=extendParamSpec
)
metadatabody = ClusterMetadata(
    name="cluster"
)
request.body = Cluster(
    spec=specbody,
    metadata=metadatabody,
    api_version="v3",
    kind="Cluster"
)
response = client.create_cluster(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)

```

- Create an HA CCE cluster of v1.25 with three master nodes and a maximum of 50 worker nodes billed on a pay-per-use basis.

```
# coding: utf-8
```

```

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

```

```
if __name__ == "__main__":
```

```

    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    environment variables and decrypted during use to ensure security.

```

```
# In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId) \

client = CceClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(CceRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = CreateClusterRequest()
    extendParamSpec = ClusterExtendParam(
        enterprise_project_id="0",
        kube_proxy_mode="iptables"
    )
    authenticationSpec = Authentication(
        mode="rbac"
    )
    serviceNetworkSpec = ServiceNetwork(
        i_pv4_cidr="10.247.0.0/16"
    )
    listSubnetsEniNetwork = [
        NetworkSubnet(
            subnet_id="861fb11d-2f0e-4c10-a98a-166dc26e4ff7"
        )
    ]
    eniNetworkSpec = EniNetwork(
        eni_subnet_id="861fb11d-2f0e-4c10-a98a-166dc26e4ff7",
        eni_subnet_cidr="192.168.0.0/24",
        subnets=listSubnetsEniNetwork
    )
    containerNetworkSpec = ContainerNetwork(
        mode="eni"
    )
    hostNetworkSpec = HostNetwork(
        vpc="030bfb19-5fa7-42ad-8a0d-c0721d268867",
        subnet="ca964acf-8468-4735-8229-97940ef6c881"
    )
    specbody = ClusterSpec(
        category="Turbo",
        type="VirtualMachine",
        flavor="cce.s2.small",
        version="v1.25",
        description="",
        ipv6enable=False,
        host_network=hostNetworkSpec,
        container_network=containerNetworkSpec,
        eni_network=eniNetworkSpec,
        service_network=serviceNetworkSpec,
        authentication=authenticationSpec,
        billing_mode=0,
        extend_param=extendParamSpec
    )
    metadatabody = ClusterMetadata(
        name="cluster"
    )
    request.body = Cluster(
        spec=specbody,
        metadata=metadatabody,
        api_version="v3",
        kind="Cluster"
    )
    response = client.create_cluster(request)
    print(response)
```

```
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

- Create an HA CCE cluster of v1.19 with three master nodes and a maximum of 50 worker nodes billed on a pay-per-use basis.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateClusterRequest{}
    enterpriseProjectIdExtendParam:= "0"
    kubeProxyModeExtendParam:= "iptables"
    alphaCceFixPoolMaskExtendParam:= "25"
    extendParamSpec := &model.ClusterExtendParam{
        EnterpriseProjectId: &enterpriseProjectIdExtendParam,
        KubeProxyMode: &kubeProxyModeExtendParam,
        AlphaCceFixPoolMask: &alphaCceFixPoolMaskExtendParam,
    }
    modeAuthentication:= "rbac"
    authenticationSpec := &model.Authentication{
        Mode: &modeAuthentication,
    }
    cidrContainerNetwork:= "10.0.0.0/16"
    containerNetworkSpec := &model.ContainerNetwork{
        Mode: model.GetContainerNetworkModeEnum().VPC_ROUTER,
        Cidr: &cidrContainerNetwork,
    }
    hostNetworkSpec := &model.HostNetwork{
        Vpc: "030bfb19-5fa7-42ad-8a0d-c0721d268867",
        Subnet: "ca964acf-8468-4735-8229-97940ef6c881",
    }
    categorySpec:= model.GetClusterSpecCategoryEnum().CCE
    versionSpec:= "v1.19"
    descriptionSpec:= ""
```



```

ipv6enableSpec:= false
billingModeSpec:= int32(0)
kubernetesSvcIpsRangeSpec:= "10.247.0.0/16"
specbody := &model.ClusterSpec{
    Category: &categorySpec,
    Flavor: "cce.s2.small",
    Version: &versionSpec,
    Description: &descriptionSpec,
    Ipv6enable: &ipv6enableSpec,
    HostNetwork: hostNetworkSpec,
    ContainerNetwork: containerNetworkSpec,
    Authentication: authenticationSpec,
    BillingMode: &billingModeSpec,
    KubernetesSvcIpsRange: &kubernetesSvcIpsRangeSpec,
    ExtendParam: extendParamSpec,
}
metadatabody := &model.ClusterMetadata{
    Name: "cluster",
}
request.Body = &model.Cluster{
    Spec: specbody,
    Metadata: metadatabody,
    ApiVersion: "v3",
    Kind: "Cluster",
}
response, err := client.CreateCluster(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}

```

- Create an HA CCE cluster of v1.19 with three master nodes and a maximum of 50 worker nodes billed on a pay-per-use basis and install ICAgent in the cluster.

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
}

```

```

request := &model.CreateClusterRequest{
enterpriseProjectIdExtendParam:= "0"
kubeProxyModeExtendParam:= "iptables"
alphaCceFixPoolMaskExtendParam:= "25"
extendParamSpec := &model.ClusterExtendParam{
    EnterpriseProjectId: &enterpriseProjectIdExtendParam,
    KubeProxyMode: &kubeProxyModeExtendParam,
    AlphaCceFixPoolMask: &alphaCceFixPoolMaskExtendParam,
}
modeAuthentication:= "rbac"
authenticationSpec := &model.Authentication{
    Mode: &modeAuthentication,
}
cidrContainerNetwork:= "10.0.0.0/16"
containerNetworkSpec := &model.ContainerNetwork{
    Mode: model.GetContainerNetworkModeEnum().VPC_ROUTER,
    Cidr: &cidrContainerNetwork,
}
hostNetworkSpec := &model.HostNetwork{
    Vpc: "030bfb19-5fa7-42ad-8a0d-c0721d268867",
    Subnet: "ca964acf-8468-4735-8229-97940ef6c881",
}
categorySpec:= model.GetClusterSpecCategoryEnum().CCE
versionSpec:= "v1.19"
descriptionSpec:= ""
ipv6enableSpec:= false
billingModeSpec:= int32(0)
kubernetesSvcIprangeSpec:= "10.247.0.0/16"
specbody := &model.ClusterSpec{
    Category: &categorySpec,
    Flavor: "cce.s2.small",
    Version: &versionSpec,
    Description: &descriptionSpec,
    Ipv6enable: &ipv6enableSpec,
    HostNetwork: hostNetworkSpec,
    ContainerNetwork: containerNetworkSpec,
    Authentication: authenticationSpec,
    BillingMode: &billingModeSpec,
    KubernetesSvcIprange: &kubernetesSvcIprangeSpec,
    ExtendParam: extendParamSpec,
}
var listAnnotationsMetadata = map[string]string{
    "cluster.install.addons.external/install": "[{"addonTemplateName":"icagent"}]",
}
metadatabody := &model.ClusterMetadata{
    Name: "cluster",
    Annotations: listAnnotationsMetadata,
}
request.Body = &model.Cluster{
    Spec: specbody,
    Metadata: metadatabody,
    ApiVersion: "v3",
    Kind: "Cluster",
}
response, err := client.CreateCluster(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

- Create an HA CCE cluster of v1.19 with three master nodes and a maximum of 50 worker nodes billed on a pay-per-use basis and configure the default security group for custom nodes in the cluster.

```
package main
```

```
import (
    "fmt"
```

```
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateClusterRequest{
        enterpriseProjectIdExtendParam:= "0"
        kubeProxyModeExtendParam:= "iptables"
        alphaCceFixPoolMaskExtendParam:= "25"
        extendParamSpec := &model.ClusterExtendParam{
            EnterpriseProjectId: &enterpriseProjectIdExtendParam,
            KubeProxyMode: &kubeProxyModeExtendParam,
            AlphaCceFixPoolMask: &alphaCceFixPoolMaskExtendParam,
        }
        modeAuthentication:= "rbac"
        authenticationSpec := &model.Authentication{
            Mode: &modeAuthentication,
        }
        cidrContainerNetwork:= "10.0.0.0/16"
        containerNetworkSpec := &model.ContainerNetwork{
            Mode: model.GetContainerNetworkModeEnum().VPC_ROUTER,
            Cidr: &cidrContainerNetwork,
        }
        securityGroupHostNetwork:= "a4ef108c-2ec6-492f-a6c4-7b64e25ae490"
        hostNetworkSpec := &model.HostNetwork{
            Vpc: "030bfb19-5fa7-42ad-8a0d-c0721d268867",
            Subnet: "ca964acf-8468-4735-8229-97940ef6c881",
            SecurityGroup: &securityGroupHostNetwork,
        }
        categorySpec:= model.GetClusterSpecCategoryEnum().CCE
        versionSpec:= "v1.19"
        descriptionSpec:= ""
        ipv6enableSpec:= false
        billingModeSpec:= int32(0)
        kubernetesSvclpRangeSpec:= "10.247.0.0/16"
        specbody := &model.ClusterSpec{
            Category: &categorySpec,
            Flavor: "cce.s2.small",
            Version: &versionSpec,
            Description: &descriptionSpec,
            Ipv6enable: &ipv6enableSpec,
            HostNetwork: hostNetworkSpec,
            ContainerNetwork: containerNetworkSpec,
            Authentication: authenticationSpec,
            BillingMode: &billingModeSpec,
        }
    }
```

```

    KubernetesSvcIpRange: &kubernetesSvcIpRangeSpec,
    ExtendParam: extendParamSpec,
  }
  metadatabody := &model.ClusterMetadata{
    Name: "cluster",
  }
  request.Body = &model.Cluster{
    Spec: specbody,
    Metadata: metadatabody,
    ApiVersion: "v3",
    Kind: "Cluster",
  }
  response, err := client.CreateCluster(request)
  if err == nil {
    fmt.Printf("%v\n", response)
  } else {
    fmt.Println(err)
  }
}

```

- Create an HA CCE cluster of v1.25 with three master nodes and a maximum of 50 worker nodes billed on a pay-per-use basis.

package main

```

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateClusterRequest{
        enterpriseProjectIdExtendParam:= "0"
        kubeProxyModeExtendParam:= "iptables"
        extendParamSpec := &model.ClusterExtendParam{
            EnterpriseProjectId: &enterpriseProjectIdExtendParam,
            KubeProxyMode: &kubeProxyModeExtendParam,
        }
        modeAuthentication:= "rbac"
        authenticationSpec := &model.Authentication{
            Mode: &modeAuthentication,
        }
        ipv4CIDRServiceNetwork:= "10.247.0.0/16"
        serviceNetworkSpec := &model.ServiceNetwork{
            IPv4CIDR: &ipv4CIDRServiceNetwork,

```

```
}
var listSubnetsEniNetwork = []model.NetworkSubnet{
    {
        SubnetID: "861fb11d-2f0e-4c10-a98a-166dc26e4ff7",
    },
}
eniSubnetCIDREniNetwork:= "192.168.0.0/24"
eniNetworkSpec := &model.EniNetwork{
    EniSubnetId: "861fb11d-2f0e-4c10-a98a-166dc26e4ff7",
    EniSubnetCIDR: &eniSubnetCIDREniNetwork,
    Subnets: listSubnetsEniNetwork,
}
containerNetworkSpec := &model.ContainerNetwork{
    Mode: model.GetContainerNetworkModeEnum().ENI,
}
hostNetworkSpec := &model.HostNetwork{
    Vpc: "030bfb19-5fa7-42ad-8a0d-c0721d268867",
    Subnet: "ca964acf-8468-4735-8229-97940ef6c881",
}
categorySpec:= model.GetClusterSpecCategoryEnum().TURBO
typeSpec:= model.GetClusterSpecTypeEnum().VIRTUAL_MACHINE
versionSpec:= "v1.25"
descriptionSpec:= ""
ipv6enableSpec:= false
billingModeSpec:= int32(0)
specbody := &model.ClusterSpec{
    Category: &categorySpec,
    Type: &typeSpec,
    Flavor: "cce.s2.small",
    Version: &versionSpec,
    Description: &descriptionSpec,
    Ipv6enable: &ipv6enableSpec,
    HostNetwork: hostNetworkSpec,
    ContainerNetwork: containerNetworkSpec,
    EniNetwork: eniNetworkSpec,
    ServiceNetwork: serviceNetworkSpec,
    Authentication: authenticationSpec,
    BillingMode: &billingModeSpec,
    ExtendParam: extendParamSpec,
}
metadatabody := &model.ClusterMetadata{
    Name: "cluster",
}
request.Body = &model.Cluster{
    Spec: specbody,
    Metadata: metadatabody,
    ApiVersion: "v3",
    Kind: "Cluster",
}
response, err := client.CreateCluster(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
201	The cluster creation job is successfully delivered.

Error Codes

See [Error Codes](#).

4.2.2 Reading a Specified Cluster

Function

This API is used to obtain details about a specified cluster.

NOTE

The URL for cluster management is in the format of `https://Endpoint/uri`. In the URL, **uri** indicates the resource path, that is, the path for API access.

Calling Method

For details, see [Calling APIs](#).

URI

GET `/api/v3/projects/{project_id}/clusters/{cluster_id}`

Table 4-40 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Table 4-41 Query Parameters

Parameter	Mandatory	Type	Description
detail	No	String	Whether the details about a cluster are queried. If this parameter is set to true , the total number of nodes (totalNodesNumber), number of normal nodes (activeNodesNumber), total CPUs (totalNodesCPU), total memory size (totalNodesMemory), and installed add-ons (installedAddonInstances) will be added to the annotation. The details of installed add-ons include the add-on name (addonTemplateName), version (version), and status (status).

Request Parameters

Table 4-42 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-43 Response body parameters

Parameter	Type	Description
kind	String	API type. The value is fixed at Cluster or cluster and cannot be changed.
apiVersion	String	API version. The value is fixed at v3 and cannot be changed.
metadata	ClusterMetadata object	Basic information about a cluster. Metadata is a collection of attributes.
spec	ClusterSpec object	Detailed description of the cluster. CCE creates or updates objects by defining or updating spec .
status	ClusterStatus object	Cluster status.

Table 4-44 ClusterMetadata

Parameter	Type	Description
name	String	Cluster name. Enter 4 to 128 characters, starting with a lowercase letter and not ending with a hyphen (-). Only lowercase letters, digits, and hyphens (-) are allowed.
uid	String	Cluster ID, which uniquely identifies a cluster. This ID is automatically generated after a cluster is created. Only the automatically generated ID will take effect. When you create a yearly/monthly cluster, no cluster ID will be returned in the response body.
alias	String	Alias of a cluster name displayed on the CCE console, and the name can be changed. Enter 4 to 128 characters, starting with a lowercase letter and not ending with a hyphen (-). Only lowercase letters, digits, and hyphens (-) are allowed. A cluster alias must be unique. In the request body for creating or updating a cluster, if the cluster alias is not specified or set to null, the cluster name will be used as the cluster alias. In the response body for obtaining a cluster, the cluster alias is returned. If it is not configured, the cluster name will be returned.

Parameter	Type	Description
annotations	Map<String,String>	<p>Cluster annotations, in the format of key-value pairs.</p> <pre>"annotations": { "key1" : "value1", "key2" : "value2" }</pre> <p>NOTE</p> <ul style="list-style-type: none"> • annotations: Does not label or select objects. The metadata in annotations may be small or large, structured or unstructured, and may include characters that are not allowed in labels. • This field is not stored in the database and is used only to specify the add-ons to be installed in the cluster. • Install ICAgent during cluster creation by adding the key-value pair "cluster.install.addons.external/install":{"addonTemplateName":"icagent"}".
labels	Map<String,String>	<p>Cluster labels, in the format of key-value pairs.</p> <p>NOTE</p> <p>The value of this field is automatically generated by the system and is used by the frontend to identify the features supported by the cluster during the upgrade. Customized values are invalid.</p>
creationTimestamp	String	Time when the cluster was created.
updateTimestamp	String	Time when the cluster was updated.

Table 4-45 ClusterSpec

Parameter	Type	Description
category	String	<p>Cluster type. Options:</p> <ul style="list-style-type: none"> • CCE: CCE cluster CCE cluster supports hybrid deployment of VMs and BMSs, and heterogeneous nodes such as GPU and NPU nodes, allowing you to run your containers in a secure and stable container runtime environment based on a high-performance network model. • Turbo: CCE Turbo cluster One-stop, cost-effective CCE Turbo clusters run on the cloud native 2.0 infrastructure featuring hardware-software synergy for lossless networking, high security and reliability, and intelligent scheduling.

Parameter	Type	Description
type	String	Master node architecture: <ul style="list-style-type: none"> • VirtualMachine: x86 • ARM64: Arm-based Kunpeng
flavor	String	Cluster specifications. Specifications of clusters 1.15 or later versions can be changed after they are created. For details, see Modifying Cluster Specifications . Options: <ul style="list-style-type: none"> • cce.s1.small: a small-scale CCE cluster with one master node and a maximum of 50 worker nodes • cce.s1.medium: a medium-scale CCE cluster with one master node and a maximum of 200 worker nodes • cce.s2.small: a small-scale CCE cluster with three master nodes and a maximum of 50 worker nodes • cce.s2.medium: a medium-scale CCE cluster with three master nodes and a maximum of 200 worker nodes • cce.s2.large: a large-scale CCE cluster with three master nodes and a maximum of 1,000 worker nodes • cce.s2.xlarge: an ultra-large-scale CCE cluster with three master nodes and a maximum of 2,000 worker nodes <p>NOTE The fields in the parameters are described as follows:</p> <ul style="list-style-type: none"> • s1: specifies a cluster with one master node. If the master node is faulty, the cluster will become unavailable, but running workloads in the cluster are not affected. • s2: specifies an HA cluster with three master nodes. If one of the master nodes is faulty, the cluster is still available. • dec: specifies a DeC CCE cluster. For example, cce.dec.s1.small specifies a small-scale, DeC CCE cluster with one master node and a maximum of 50 worker nodes. • small: specifies that a cluster can manage a maximum of 50 worker nodes. • medium: specifies that a cluster can manage a maximum of 200 worker nodes. • large: specifies that a cluster can manage a maximum of 1,000 worker nodes. • xlarge: specifies that a cluster can manage a maximum of 2,000 worker nodes.

Parameter	Type	Description
version	String	<p>Cluster version, which mirrors the baseline version of the Kubernetes community. The latest version is recommended.</p> <p>You can create clusters of two latest versions on the CCE console. To learn which cluster versions are available, log in to the CCE console, create a cluster, and check the Cluster Version parameter.</p> <p>You can call APIs to create clusters of other versions. However, these cluster versions will be gradually terminated. For details about the support policy, see the CCE announcement.</p> <p>NOTE</p> <ul style="list-style-type: none"> • If not specified, a cluster of the latest version will be created. • If a baseline cluster version is specified but the R version is not specified, a cluster of the latest R version will be created by default. It is a good practice not to specify the R version. • CCE Turbo clusters of v1.19 or later are commercially available.
platformVersion	String	<p>CCE cluster platform version, indicating the internal version under the cluster version (version). Platform versions are used to trace iterations in a major cluster version. They are unique within a major cluster version and recounted when the major cluster version changes. This parameter cannot be customized. When you create a cluster, the latest corresponding platform version is automatically selected.</p> <p>The format of platformVersion is cce.X.Y.</p> <ul style="list-style-type: none"> • X: internal feature version, indicating changes in features, patches, or OS support in the cluster version. The value starts from 1 and increases monotonically. • Y: patch version of an internal feature version. It is used only for software package update after the feature version is brought online. No other modification is involved. The value starts from 0 and increases monotonically.

Parameter	Type	Description
description	String	Cluster description, for example, which purpose the cluster is intended to serve. By default, this field is left unspecified. To modify cluster description after the cluster is created, call the API for updating cluster information or go to the cluster details page on the CCE console. Only UTF-8 encoding is supported.
customSan	Array of strings	Custom SAN field in the server certificate of the cluster API server, which must comply with the SSL and X509 format specifications. <ol style="list-style-type: none"> 1. Duplicate names are not allowed. 2. Must comply with the IP address and domain name formats. Example: SAN 1: DNS Name=example.com SAN 2: DNS Name=www.example.com SAN 3: DNS Name=example.net SAN 4: IP Address=93.184.216.34
ipv6enable	Boolean	Whether the cluster supports IPv6 addresses. This field is supported in clusters of v1.15 and later versions.
hostNetwork	HostNetwork object	Node networking parameters, including VPC and subnet ID. This field is mandatory because nodes in a cluster communicate with each other by using a VPC.
containerNetwork	ContainerNetwork object	Container networking parameters, including the container network model and container CIDR block.
eniNetwork	EniNetwork object	Configuration of the Cloud Native Network 2.0 model. Specify this field when creating a CCE Turbo cluster.
serviceNetwork	ServiceNetwork object	Service CIDR block, including IPv4 CIDR blocks.
authentication	Authentication object	Configurations of the cluster authentication mode.
billingMode	Integer	Billing mode of a cluster. <ul style="list-style-type: none"> • 0: pay-per-use • 1: yearly/monthly Defaults to pay-per-use.
masters	Array of MasterSpec objects	Advanced configurations of master nodes

Parameter	Type	Description
kubernetesSvcIpRange	String	Service CIDR blocks for Kubernetes clusterIPs. This field is available only for clusters of v1.11.7 and later. If this parameter is not specified during cluster creation, the default value 10.247.0.0/16 will be used. This parameter is deprecated. Use serviceNetwork instead. The new field contains the IPv4 CIDR blocks.
clusterTags	Array of ResourceTag objects	Cluster resource tags.
kubeProxyMode	String	Service forwarding mode. Options: <ul style="list-style-type: none"> • iptables: Traditional kube-proxy uses iptables rules to implement Service load balancing. In this mode, too many iptables rules will be generated when many Services are deployed. In addition, non-incremental updates will cause latency and even tangible performance issues in the case of service traffic spikes. • ipvs: Optimized kube-proxy mode with higher throughput and faster speed. This mode supports incremental updates and can keep connections uninterrupted during Service updates. It is suitable for large-sized clusters. <p>NOTE iptables is used by default.</p>
az	String	AZ. This field is returned only for a query. For details about AZs supported by CCE, see Regions and Endpoints .
extendParam	ClusterExtendedParam object	Extended field to decide whether the cluster will span across AZs or belong to a specified enterprise project, or whether a dedicated CCE cluster is to be created.
supportIstio	Boolean	Whether Istio is supported.
configurationsOverride	Array of PackageConfiguration objects	Cluster default component configuration override. If you specify a component or parameter that is not supported, the configuration item will be ignored. For details about the supported components and their parameters, see Cluster Configuration Management .

Table 4-46 HostNetwork

Parameter	Type	Description
vpc	String	<p>ID of the VPC used to create a master node. You can obtain it in either of the following ways:</p> <ul style="list-style-type: none"> • Method 1: Log in to the VPC console and view the VPC ID in the VPC details page. • Method 2: Use the VPC API to obtain VPC IDs. <p>For details, see Querying VPCs.</p>
subnet	String	<p>Network ID of the subnet used to create a master node. You can obtain it in either of the following ways:</p> <ul style="list-style-type: none"> • Method 1: Log in to the VPC console and click the target subnet on the Subnets page. You can view the network ID on the displayed page. • Method 2: Use the VPC API to obtain subnets. <p>For details, see Querying Subnets.</p>
SecurityGroup	String	<p>Default worker node security group ID of the cluster. If specified, the cluster will be bound to the target security group. Otherwise, the system will automatically create a default worker node security group for you. The default worker node security group needs to allow access from certain ports to ensure normal communications. For details, see How Do I Harden the Automatically Created Security Group Rules for CCE Cluster Nodes.</p>

Table 4-47 ContainerNetwork

Parameter	Type	Description
mode	String	<p>Container network model. Select one of the following possible values:</p> <ul style="list-style-type: none"> • overlay_l2: an overlay_l2 network (container tunnel network) built for containers by using OpenVSwitch (OVS). • vpc-router: an underlay_l2 network built for containers by using IPvlan and custom VPC routes. • eni: Cloud Native Network 2.0. This model has integrated cloud native elastic network interfaces (ENIs), uses VPC CIDR blocks to allocate container IP addresses, and allows direct traffic distribution to containers through a load balancer for high performance. Use this model when creating a CCE Turbo cluster.
cidr	String	<p>Container CIDR block. Recommended: 10.0.0.0/12-19, 172.16.0.0/16-19, or 192.168.0.0/16-19. If the selected CIDR block conflicts with existing ones, an error will be reported.</p> <p>Not editable after the cluster is created. (deprecated. A specified cidrs will make cidr invalid.)</p>
cidrs	Array of ContainerCIDR objects	<p>List of container CIDR blocks. In clusters of v1.21 or later, the cidrs field is used. When the cluster network type is vpc-router, you can configure a maximum of 20 container CIDR blocks. In cluster versions earlier than v1.21, if the cidrs field is used, the first CIDR element in the array is used as the container CIDR block.</p> <p>The configuration cannot be modified after the cluster is created.</p>

Table 4-48 ContainerCIDR

Parameter	Type	Description
cidr	String	<p>Container CIDR block. Recommended: 10.0.0.0/12-19, 172.16.0.0/16-19, and 192.168.0.0/16-19</p>

Table 4-49 EniNetwork

Parameter	Type	Description
eniSubnetId	String	IPv4 subnet ID of ENI subnet. (IPv6 is not supported and is being discarded.) You can obtain it in either of the following ways: <ul style="list-style-type: none"> Method 1: Log in to the VPC console and click the target subnet on the Subnets page. You can view the IPv4 subnet ID on the displayed page. Method 2: Use the VPC API to obtain subnet IDs. For details, see Querying Subnets .
eniSubnetCIDR	String	ENI subnet CIDR (being discarded)
subnets	Array of NetworkSubnet objects	List of IPv4 subnet IDs

Table 4-50 NetworkSubnet

Parameter	Type	Description
subnetID	String	IPv4 subnet ID of the subnet for creating master nodes. IPv6 is not supported. You can obtain it in either of the following ways: <ul style="list-style-type: none"> Method 1: Log in to the VPC console and click the target subnet on the Subnets page. You can view the IPv4 subnet ID on the displayed page. Method 2: Use the VPC API to obtain subnets. For details, see Querying Subnets .

Table 4-51 ServiceNetwork

Parameter	Type	Description
IPv4CIDR	String	Value range of the Kubernetes clusterIP IPv4 CIDR blocks. If this parameter is not specified during cluster creation, the default value 10.247.0.0/16 will be used.

Table 4-52 Authentication

Parameter	Type	Description
mode	String	Cluster authentication mode. <ul style="list-style-type: none"> Clusters of Kubernetes v1.11 or earlier support x509, rbac, and authenticating_proxy. Defaults to x509. Clusters of Kubernetes v1.13 or later support rbac and authenticating_proxy. Defaults to rbac.
authenticatingProxy	AuthenticatingProxy object	Configuration related to the <code>authenticating_proxy</code> mode. This field is mandatory when the authentication mode is authenticating_proxy .

Table 4-53 AuthenticatingProxy

Parameter	Type	Description
ca	String	X509 CA certificate (Base64-encoded) configured in <code>authenticating_proxy</code> mode. This field is mandatory when the cluster authentication mode is authenticating_proxy . Maximum size: 1 MB
cert	String	Client certificate issued by the X509 CA certificate configured in <code>authenticating_proxy</code> mode, which is used for authentication from kube-apiserver to the extended API server. (The value must be Base64-encoded.) This field is mandatory when the cluster authentication mode is authenticating_proxy .
privateKey	String	Private key of the client certificate issued by the X509 CA certificate configured in <code>authenticating_proxy</code> mode, which is used for authentication from kube-apiserver to the extended API server. The private key used by the Kubernetes cluster does not support password encryption. Use an unencrypted private key. (The value must be Base64-encoded.) This field is mandatory when the cluster authentication mode is authenticating_proxy .

Table 4-54 MasterSpec

Parameter	Type	Description
availabilityZone	String	AZ

Table 4-55 ResourceTag

Parameter	Type	Description
key	String	Key. <ul style="list-style-type: none"> • Cannot be null. Max characters: 128. • Use letters, digits, and spaces in UTF-8 format. • Can contain the following special characters: <code>._:/=+-@</code>. • Cannot start with <code>_sys_</code>.
value	String	Value. <ul style="list-style-type: none"> • Can be null but not the default. Max characters: 255. • Use letters, digits, and spaces in UTF-8 format. • Can contain the following special characters: <code>._:/=+-@</code>.

Table 4-56 ClusterExtendParam

Parameter	Type	Description
clusterAZ	String	AZ of master nodes in a cluster. For details about AZs supported by CCE, see Regions and Endpoints . <ul style="list-style-type: none"> • multi_az: (Optional) The cluster will span across AZs. Multiple AZs can be configured only when a cluster with multiple master nodes is used. • <i>AZ of the dedicated cloud computing pool</i>: The cluster will be deployed in the DeC AZ. It is mandatory for DeC CCE clusters.

Parameter	Type	Description
dssMasterVolumes	String	<p>Whether the system and data disks of a master node use dedicated distributed storage. If this parameter is omitted or left unspecified, EVS disks are used by default.</p> <p>This parameter is mandatory for dedicated CCE clusters. It is in the following format: <code><rootVol.dssPoolID>.<rootVol.volType>;<dataVol.dssPoolID>.<dataVol.volType></code></p> <p>Field description:</p> <ul style="list-style-type: none"> • rootVol is the system disk. dataVol is the data disk. • dssPoolID indicates the ID of the DSS storage pool. • volType indicates the storage volume type of the DSS storage pool, such as SAS and SSD. <p>Example: <code>c950ee97-587c-4f24-8a74-3367e3da570f.sas;6edbc2f4-1507-44f8-ac0d-eed1d2608d38.ssd</code></p> <p>NOTE This field cannot be configured for non-dedicated CCE clusters.</p>
enterpriseProjectId	String	<p>ID of the enterprise project that a cluster belongs to.</p> <p>NOTE</p> <ul style="list-style-type: none"> • An enterprise project can be configured only after the enterprise project function is enabled. • The enterprise project to which the cluster belongs must be the same as that to which other cloud service resources associated with the cluster belong.

Parameter	Type	Description
kubeProxyMode	String	<p>Service forwarding mode. Two modes are available:</p> <ul style="list-style-type: none"> • iptables: Traditional kube-proxy uses iptables rules to implement Service load balancing. In this mode, too many iptables rules will be generated when many Services are deployed. In addition, non-incremental updates will cause latency and even tangible performance issues in the case of service traffic spikes. • ipvs: Optimized kube-proxy mode with higher throughput and faster speed. This mode supports incremental updates and can keep connections uninterrupted during Service updates. It is suitable for large-sized clusters. <p>NOTE This parameter has been deprecated. If this parameter and kubeProxyMode in ClusterSpec are specified at the same time, the latter is used.</p>
clusterExternalIP	String	EIP of the master node
alpha.cce/fixPoolMask	String	<p>Number of mask bits of the fixed IP address pool of the container network model. This field is supported only for the VPC network model (vpc-router).</p> <p>This parameter determines the number of container IP addresses that can be allocated to a node. The maximum number of pods that can be created on a node is decided by this parameter and maxPods set during node creation.</p> <p>For details, see Maximum Number of Pods That Can Be Created on a Node.</p> <p>For integer characters, the value ranges from 24 to 28.</p>
decMasterFlavor	String	Specifications of the master node in the dedicated hybrid cluster.
dockerUmaskMode	String	Default UmaskMode configuration of Docker in a cluster. The value can be secure or normal . If this parameter is not specified, normal is used by default.

Parameter	Type	Description
kubernetes.io/cpuManagerPolicy	String	<p>Cluster CPU management policy. The value can be none (or null) or static. The default value is none (or null).</p> <ul style="list-style-type: none"> ● none or null: disables pods from exclusively occupying CPUs. Select this option if you want a large pool of shareable CPU cores. ● static: enables pods to exclusively occupy CPUs. Select this option if your workload is sensitive to CPU cache and scheduling latency. In a CCE Turbo cluster, this setting is valid only for nodes where common containers, not Kata containers, run.
orderId	String	<p>Order ID. This parameter is returned in the response when the cluster is billed on a yearly/monthly basis with auto payment enabled (only in creation scenarios).</p>
periodType	String	<ul style="list-style-type: none"> ● month: The unit is month. ● year: The unit is year. <p>NOTE Request parameter, which is valid and mandatory when billingMode is set to 1 (yearly/monthly billing). Response parameter, which is returned only when a yearly/monthly cluster is created.</p>
periodNum	Integer	<p>Subscription duration. The value can be:</p> <ul style="list-style-type: none"> ● If periodType is month, the value ranges from 1 to 9. ● If periodType is year, the value ranges from 1 to 3. <p>NOTE Request parameter, which is valid and mandatory when billingMode is set to 1. Response parameter, which is returned only when a yearly/monthly cluster is created.</p>
isAutoRenew	String	<p>Whether auto renewal is enabled.</p> <ul style="list-style-type: none"> ● true: Auto renewal is enabled. ● false: Auto renewal is not enabled. <p>NOTE This field is valid when billingMode is set to 1. If not specified, auto renewal is not enabled.</p>

Parameter	Type	Description
isAutoPay	String	Whether to deduct fees automatically. <ul style="list-style-type: none"> • true: Enable automatic fee deduction. • false: Do not enable automatic fee deduction. <p>NOTE This field is valid when billingMode is set to 1. If not specified, auto fee deduction is not enabled.</p>
upgradefrom	String	Records of how the cluster is upgraded to the current version

Table 4-57 PackageConfiguration

Parameter	Type	Description
name	String	Component name.
configurations	Array of ConfigurationItem objects	Component configuration item.

Table 4-58 ConfigurationItem

Parameter	Type	Description
name	String	Component configuration item name.
value	Object	Component configuration item value.

Table 4-59 ClusterStatus

Parameter	Type	Description
phase	String	<p>Cluster status. Options:</p> <ul style="list-style-type: none"> ● Available: The cluster is running properly. ● Unavailable: The cluster is exhibiting unexpected behavior. Manually delete it. ● ScalingUp: Nodes are being added to the cluster. ● ScalingDown: The cluster is being downsized to fewer nodes. ● Creating: The cluster is being created. ● Deleting: The cluster is being deleted. ● Upgrading: The cluster is being upgraded. ● Resizing: Cluster specifications are being changed. ● RollingBack: The cluster is being rolled back. ● RollbackFailed: The cluster rollback is abnormal. ● Hibernating: The cluster is being hibernated. ● Hibernation: The cluster is in hibernation. ● Awaking: The cluster is being woken up from hibernation. ● Empty: The cluster does not have any resources. This field is discarded. ● Error: Resources in the cluster are abnormal. Manually delete the cluster.
jobID	String	<p>ID of the task associated with the cluster in the current state. Options:</p> <ul style="list-style-type: none"> ● ID of the associated task returned when creating a cluster. You can use it to obtain the auxiliary tasks for creating a cluster. ● ID of the associated task returned when a cluster fails to be deleted or is deleted. If this parameter is not empty, you can use the task ID to obtain the auxiliary tasks for deleting a cluster. <p>NOTE Tasks are short-lived. Do not use task information in scenarios such as cluster status determination.</p>
reason	String	Reason of cluster state change. This parameter is returned if the cluster is not in the Available state.

Parameter	Type	Description
message	String	Detailed information about why the cluster changes to the current state. This parameter is returned if the cluster is not in the Available state.
endpoints	Array of ClusterEndpoints objects	Access address of kube-apiserver in the cluster.
isLocked	Boolean	The CBC resource is locked.
lockScene	String	Scenario where the CBC resource is locked.
lockSource	String	Resource locking.
lockSourceId	String	ID of the locked resource.
deleteOption	Object	Whether to delete configurations. This parameter is contained only in the response to the deletion request.
deleteStatus	Object	Whether to delete the status information. This parameter is contained only in the response to the deletion request.

Table 4-60 ClusterEndpoints

Parameter	Type	Description
url	String	Access address of kube-apiserver in the cluster.
type	String	Type of the cluster access address. <ul style="list-style-type: none"> • Internal: address for internal network access • External: address for external network access

Example Requests

None

Example Responses

Status code: 200

Information about the specified cluster is successfully obtained.

```
{
  "kind": "Cluster",
  "apiVersion": "v3",
  "metadata": {
```



```

"alias" : "mycluster",
"name" : "mycluster",
"uid" : "4d1ecb2c-229a-11e8-9c75-0255ac100ceb",
"creationTimestamp" : "2018-08-02 03:48:58.968214406 +0000 UTC",
"updateTimestamp" : "2018-08-02 04:05:29.386391813 +0000 UTC"
},
"spec" : {
  "type" : "VirtualMachine",
  "flavor" : "cce.s1.small",
  "version" : "v1.7.3-r13",
  "description" : "this is a demo cluster",
  "customSan" : [ "192.168.1.0", "example.com" ],
  "hostNetwork" : {
    "vpc" : "4d1ecb2c-229a-11e8-9c75-0255ac100ceb",
    "subnet" : "4d1ecb2c-229a-11e8-9c75-0255ac100ceb"
  },
  "containerNetwork" : {
    "mode" : "overlay_l2",
    "cidr" : "172.16.0.0/16"
  },
  "authentication" : {
    "mode" : "x509",
    "authenticatingProxy" : { }
  },
  "billingMode" : 0
},
"status" : {
  "phase" : "Available",
  "endpoints" : [ {
    "url" : "https://192.168.0.11:5443",
    "type" : "Internal"
  } ]
}
}
}

```

SDK Sample Code

The SDK sample code is as follows.

Java

```

package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ShowClusterSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)

```

```
        .withSk(sk);

    CceClient client = CceClient.newBuilder()
        .withCredential(auth)
        .withRegion(CceRegion.valueOf("<YOUR REGION>"))
        .build();
    ShowClusterRequest request = new ShowClusterRequest();
    request.withClusterId("{cluster_id}");
    try {
        ShowClusterResponse response = client.showCluster(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowClusterRequest()
        request.cluster_id = "{cluster_id}"
        response = client.show_cluster(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
```

```

cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowClusterRequest{}
    request.ClusterId = "{cluster_id}"
    response, err := client.ShowCluster(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Information about the specified cluster is successfully obtained.

Error Codes

See [Error Codes](#).

4.2.3 Listing Clusters in a Specified Project

Function

This API is used to obtain details about all clusters in a specified project.

Calling Method

For details, see [Calling APIs](#).

URI

GET /api/v3/projects/{project_id}/clusters

Table 4-61 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Table 4-62 Query Parameters

Parameter	Mandatory	Type	Description
detail	No	String	Whether the details about a cluster are queried. If this parameter is set to true , the total number of nodes (totalNodesNumber), number of normal nodes (activeNodesNumber), total CPUs (totalNodesCPU), total memory size (totalNodesMemory), and installed add-ons (installedAddonInstances) will be added to the annotation. The details of installed add-ons include the add-on name (addonTemplateName), version (version), and status (status).

Parameter	Mandatory	Type	Description
status	No	String	<p>Cluster status. Options:</p> <ul style="list-style-type: none"> • Available: The cluster is running properly. • Unavailable: The cluster is exhibiting unexpected behavior. Manually delete it. • ScalingUp: Nodes are being added to the cluster. • ScalingDown: The cluster is being downsized to fewer nodes. • Creating: The cluster is being created. • Deleting: The cluster is being deleted. • Upgrading: The cluster is being upgraded. • Resizing: Cluster specifications are being changed. • RollingBack: The cluster is being rolled back. • RollbackFailed: The cluster rollback is abnormal. • Hibernating: The cluster is being hibernated. • Hibernation: The cluster is in hibernation. • Awaking: The cluster is being woken up from hibernation. • Empty: The cluster does not have any resources. This field is discarded. • Error: Resources in the cluster are abnormal. Manually delete the cluster.
type	No	String	<p>Cluster type. Options:</p> <ul style="list-style-type: none"> • VirtualMachine: CCE cluster • ARM64: Kunpeng cluster
version	No	String	Cluster version filtering.

Request Parameters

Table 4-63 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-64 Response body parameters

Parameter	Type	Description
kind	String	Api type
apiVersion	String	API version
items	Array of Cluster objects	A list of details for all clusters in the current project. You can filter clusters by items.metadata.name .

Table 4-65 Cluster

Parameter	Type	Description
kind	String	API type. The value is fixed at Cluster or cluster and cannot be changed.
apiVersion	String	API version. The value is fixed at v3 and cannot be changed.
metadata	ClusterMetadata object	Basic information about a cluster. Metadata is a collection of attributes.

Parameter	Type	Description
spec	ClusterSpec object	Detailed description of the cluster. CCE creates or updates objects by defining or updating spec .
status	ClusterStatus object	Cluster status and job ID of the cluster creation job.

Table 4-66 ClusterMetadata

Parameter	Type	Description
name	String	Cluster name. Enter 4 to 128 characters, starting with a lowercase letter and not ending with a hyphen (-). Only lowercase letters, digits, and hyphens (-) are allowed.
uid	String	Cluster ID, which uniquely identifies a cluster. This ID is automatically generated after a cluster is created. Only the automatically generated ID will take effect. When you create a yearly/monthly cluster, no cluster ID will be returned in the response body.
alias	String	Alias of a cluster name displayed on the CCE console, and the name can be changed. Enter 4 to 128 characters, starting with a lowercase letter and not ending with a hyphen (-). Only lowercase letters, digits, and hyphens (-) are allowed. A cluster alias must be unique. In the request body for creating or updating a cluster, if the cluster alias is not specified or set to null, the cluster name will be used as the cluster alias. In the response body for obtaining a cluster, the cluster alias is returned. If it is not configured, the cluster name will be returned.

Parameter	Type	Description
annotations	Map<String,String>	<p>Cluster annotations, in the format of key-value pairs.</p> <pre>"annotations": { "key1" : "value1", "key2" : "value2" }</pre> <p>NOTE</p> <ul style="list-style-type: none"> • annotations: Does not label or select objects. The metadata in annotations may be small or large, structured or unstructured, and may include characters that are not allowed in labels. • This field is not stored in the database and is used only to specify the add-ons to be installed in the cluster. • Install ICAgent during cluster creation by adding the key-value pair "cluster.install.addons.external/install":{"addonTemplateName":"icagent"}".
labels	Map<String,String>	<p>Cluster labels, in the format of key-value pairs.</p> <p>NOTE</p> <p>The value of this field is automatically generated by the system and is used by the frontend to identify the features supported by the cluster during the upgrade. Customized values are invalid.</p>
creationTimestamp	String	Time when the cluster was created.
updateTimestamp	String	Time when the cluster was updated.

Table 4-67 ClusterSpec

Parameter	Type	Description
category	String	<p>Cluster type. Options:</p> <ul style="list-style-type: none"> • CCE: CCE cluster CCE cluster supports hybrid deployment of VMs and BMSs, and heterogeneous nodes such as GPU and NPU nodes, allowing you to run your containers in a secure and stable container runtime environment based on a high-performance network model. • Turbo: CCE Turbo cluster One-stop, cost-effective CCE Turbo clusters run on the cloud native 2.0 infrastructure featuring hardware-software synergy for lossless networking, high security and reliability, and intelligent scheduling.

Parameter	Type	Description
type	String	Master node architecture: <ul style="list-style-type: none"> • VirtualMachine: x86 • ARM64: Arm-based Kunpeng
flavor	String	Cluster specifications. Specifications of clusters 1.15 or later versions can be changed after they are created. For details, see Modifying Cluster Specifications . Options: <ul style="list-style-type: none"> • cce.s1.small: a small-scale CCE cluster with one master node and a maximum of 50 worker nodes • cce.s1.medium: a medium-scale CCE cluster with one master node and a maximum of 200 worker nodes • cce.s2.small: a small-scale CCE cluster with three master nodes and a maximum of 50 worker nodes • cce.s2.medium: a medium-scale CCE cluster with three master nodes and a maximum of 200 worker nodes • cce.s2.large: a large-scale CCE cluster with three master nodes and a maximum of 1,000 worker nodes • cce.s2.xlarge: an ultra-large-scale CCE cluster with three master nodes and a maximum of 2,000 worker nodes <p>NOTE The fields in the parameters are described as follows:</p> <ul style="list-style-type: none"> • s1: specifies a cluster with one master node. If the master node is faulty, the cluster will become unavailable, but running workloads in the cluster are not affected. • s2: specifies an HA cluster with three master nodes. If one of the master nodes is faulty, the cluster is still available. • dec: specifies a DeC CCE cluster. For example, cce.dec.s1.small specifies a small-scale, DeC CCE cluster with one master node and a maximum of 50 worker nodes. • small: specifies that a cluster can manage a maximum of 50 worker nodes. • medium: specifies that a cluster can manage a maximum of 200 worker nodes. • large: specifies that a cluster can manage a maximum of 1,000 worker nodes. • xlarge: specifies that a cluster can manage a maximum of 2,000 worker nodes.

Parameter	Type	Description
version	String	<p>Cluster version, which mirrors the baseline version of the Kubernetes community. The latest version is recommended.</p> <p>You can create clusters of two latest versions on the CCE console. To learn which cluster versions are available, log in to the CCE console, create a cluster, and check the Cluster Version parameter.</p> <p>You can call APIs to create clusters of other versions. However, these cluster versions will be gradually terminated. For details about the support policy, see the CCE announcement.</p> <p>NOTE</p> <ul style="list-style-type: none"> • If not specified, a cluster of the latest version will be created. • If a baseline cluster version is specified but the R version is not specified, a cluster of the latest R version will be created by default. It is a good practice not to specify the R version. • CCE Turbo clusters of v1.19 or later are commercially available.
platformVersion	String	<p>CCE cluster platform version, indicating the internal version under the cluster version (version). Platform versions are used to trace iterations in a major cluster version. They are unique within a major cluster version and recounted when the major cluster version changes. This parameter cannot be customized. When you create a cluster, the latest corresponding platform version is automatically selected.</p> <p>The format of platformVersion is cce.X.Y.</p> <ul style="list-style-type: none"> • X: internal feature version, indicating changes in features, patches, or OS support in the cluster version. The value starts from 1 and increases monotonically. • Y: patch version of an internal feature version. It is used only for software package update after the feature version is brought online. No other modification is involved. The value starts from 0 and increases monotonically.

Parameter	Type	Description
description	String	Cluster description, for example, which purpose the cluster is intended to serve. By default, this field is left unspecified. To modify cluster description after the cluster is created, call the API for updating cluster information or go to the cluster details page on the CCE console. Only UTF-8 encoding is supported.
customSan	Array of strings	Custom SAN field in the server certificate of the cluster API server, which must comply with the SSL and X509 format specifications. <ol style="list-style-type: none"> 1. Duplicate names are not allowed. 2. Must comply with the IP address and domain name formats. <p>Example:</p> <pre>SAN 1: DNS Name=example.com SAN 2: DNS Name=www.example.com SAN 3: DNS Name=example.net SAN 4: IP Address=93.184.216.34</pre>
ipv6enable	Boolean	Whether the cluster supports IPv6 addresses. This field is supported in clusters of v1.15 and later versions.
hostNetwork	HostNetwork object	Node networking parameters, including VPC and subnet ID. This field is mandatory because nodes in a cluster communicate with each other by using a VPC.
containerNetwork	ContainerNetwork object	Container networking parameters, including the container network model and container CIDR block.
eniNetwork	EniNetwork object	Configuration of the Cloud Native Network 2.0 model. Specify this field when creating a CCE Turbo cluster.
serviceNetwork	ServiceNetwork object	Service CIDR block, including IPv4 CIDR blocks.
authentication	Authentication object	Configurations of the cluster authentication mode.
billingMode	Integer	Billing mode of a cluster. <ul style="list-style-type: none"> • 0: pay-per-use • 1: yearly/monthly Defaults to pay-per-use.
masters	Array of MasterSpec objects	Advanced configurations of master nodes

Parameter	Type	Description
kubernetesSvcIpRange	String	Service CIDR blocks for Kubernetes clusterIPs. This field is available only for clusters of v1.11.7 and later. If this parameter is not specified during cluster creation, the default value 10.247.0.0/16 will be used. This parameter is deprecated. Use serviceNetwork instead. The new field contains the IPv4 CIDR blocks.
clusterTags	Array of ResourceTag objects	Cluster resource tags.
kubeProxyMode	String	Service forwarding mode. Options: <ul style="list-style-type: none"> • iptables: Traditional kube-proxy uses iptables rules to implement Service load balancing. In this mode, too many iptables rules will be generated when many Services are deployed. In addition, non-incremental updates will cause latency and even tangible performance issues in the case of service traffic spikes. • ipvs: Optimized kube-proxy mode with higher throughput and faster speed. This mode supports incremental updates and can keep connections uninterrupted during Service updates. It is suitable for large-sized clusters. <p>NOTE iptables is used by default.</p>
az	String	AZ. This field is returned only for a query. For details about AZs supported by CCE, see Regions and Endpoints .
extendParam	ClusterExtendedParam object	Extended field to decide whether the cluster will span across AZs or belong to a specified enterprise project, or whether a dedicated CCE cluster is to be created.
supportIstio	Boolean	Whether Istio is supported.
configurationsOverride	Array of PackageConfiguration objects	Cluster default component configuration override. If you specify a component or parameter that is not supported, the configuration item will be ignored. For details about the supported components and their parameters, see Cluster Configuration Management .

Table 4-68 HostNetwork

Parameter	Type	Description
vpc	String	<p>ID of the VPC used to create a master node. You can obtain it in either of the following ways:</p> <ul style="list-style-type: none"> Method 1: Log in to the VPC console and view the VPC ID in the VPC details page. Method 2: Use the VPC API to obtain VPC IDs. <p>For details, see Querying VPCs.</p>
subnet	String	<p>Network ID of the subnet used to create a master node. You can obtain it in either of the following ways:</p> <ul style="list-style-type: none"> Method 1: Log in to the VPC console and click the target subnet on the Subnets page. You can view the network ID on the displayed page. Method 2: Use the VPC API to obtain subnets. <p>For details, see Querying Subnets.</p>
SecurityGroup	String	<p>Default worker node security group ID of the cluster. If specified, the cluster will be bound to the target security group. Otherwise, the system will automatically create a default worker node security group for you. The default worker node security group needs to allow access from certain ports to ensure normal communications. For details, see How Do I Harden the Automatically Created Security Group Rules for CCE Cluster Nodes.</p>

Table 4-69 ContainerNetwork

Parameter	Type	Description
mode	String	<p>Container network model. Select one of the following possible values:</p> <ul style="list-style-type: none"> • overlay_l2: an overlay_l2 network (container tunnel network) built for containers by using OpenVSwitch (OVS). • vpc-router: an underlay_l2 network built for containers by using IPvlan and custom VPC routes. • eni: Cloud Native Network 2.0. This model has integrated cloud native elastic network interfaces (ENIs), uses VPC CIDR blocks to allocate container IP addresses, and allows direct traffic distribution to containers through a load balancer for high performance. Use this model when creating a CCE Turbo cluster.
cidr	String	<p>Container CIDR block. Recommended: 10.0.0.0/12-19, 172.16.0.0/16-19, or 192.168.0.0/16-19. If the selected CIDR block conflicts with existing ones, an error will be reported.</p> <p>Not editable after the cluster is created. (deprecated. A specified cidrs will make cidr invalid.)</p>
cidrs	Array of ContainerCIDR objects	<p>List of container CIDR blocks. In clusters of v1.21 or later, the cidrs field is used. When the cluster network type is vpc-router, you can configure a maximum of 20 container CIDR blocks. In cluster versions earlier than v1.21, if the cidrs field is used, the first CIDR element in the array is used as the container CIDR block.</p> <p>The configuration cannot be modified after the cluster is created.</p>

Table 4-70 ContainerCIDR

Parameter	Type	Description
cidr	String	<p>Container CIDR block. Recommended: 10.0.0.0/12-19, 172.16.0.0/16-19, and 192.168.0.0/16-19</p>

Table 4-71 EniNetwork

Parameter	Type	Description
eniSubnetId	String	IPv4 subnet ID of ENI subnet. (IPv6 is not supported and is being discarded.) You can obtain it in either of the following ways: <ul style="list-style-type: none"> Method 1: Log in to the VPC console and click the target subnet on the Subnets page. You can view the IPv4 subnet ID on the displayed page. Method 2: Use the VPC API to obtain subnet IDs. For details, see Querying Subnets .
eniSubnetCIDR	String	ENI subnet CIDR (being discarded)
subnets	Array of NetworkSubnet objects	List of IPv4 subnet IDs

Table 4-72 NetworkSubnet

Parameter	Type	Description
subnetID	String	IPv4 subnet ID of the subnet for creating master nodes. IPv6 is not supported. You can obtain it in either of the following ways: <ul style="list-style-type: none"> Method 1: Log in to the VPC console and click the target subnet on the Subnets page. You can view the IPv4 subnet ID on the displayed page. Method 2: Use the VPC API to obtain subnets. For details, see Querying Subnets .

Table 4-73 ServiceNetwork

Parameter	Type	Description
IPv4CIDR	String	Value range of the Kubernetes clusterIP IPv4 CIDR blocks. If this parameter is not specified during cluster creation, the default value 10.247.0.0/16 will be used.

Table 4-74 Authentication

Parameter	Type	Description
mode	String	Cluster authentication mode. <ul style="list-style-type: none"> Clusters of Kubernetes v1.11 or earlier support x509, rbac, and authenticating_proxy. Defaults to x509. Clusters of Kubernetes v1.13 or later support rbac and authenticating_proxy. Defaults to rbac.
authenticatingProxy	AuthenticatingProxy object	Configuration related to the <code>authenticating_proxy</code> mode. This field is mandatory when the authentication mode is authenticating_proxy .

Table 4-75 AuthenticatingProxy

Parameter	Type	Description
ca	String	X509 CA certificate (Base64-encoded) configured in <code>authenticating_proxy</code> mode. This field is mandatory when the cluster authentication mode is authenticating_proxy . Maximum size: 1 MB
cert	String	Client certificate issued by the X509 CA certificate configured in <code>authenticating_proxy</code> mode, which is used for authentication from kube-apiserver to the extended API server. (The value must be Base64-encoded.) This field is mandatory when the cluster authentication mode is authenticating_proxy .
privateKey	String	Private key of the client certificate issued by the X509 CA certificate configured in <code>authenticating_proxy</code> mode, which is used for authentication from kube-apiserver to the extended API server. The private key used by the Kubernetes cluster does not support password encryption. Use an unencrypted private key. (The value must be Base64-encoded.) This field is mandatory when the cluster authentication mode is authenticating_proxy .

Table 4-76 MasterSpec

Parameter	Type	Description
availabilityZone	String	AZ

Table 4-77 ResourceTag

Parameter	Type	Description
key	String	Key. <ul style="list-style-type: none"> Cannot be null. Max characters: 128. Use letters, digits, and spaces in UTF-8 format. Can contain the following special characters: <code>._:/=-@</code>. Cannot start with <code>_sys_</code>.
value	String	Value. <ul style="list-style-type: none"> Can be null but not the default. Max characters: 255. Use letters, digits, and spaces in UTF-8 format. Can contain the following special characters: <code>._:/=-@</code>.

Table 4-78 ClusterExtendParam

Parameter	Type	Description
clusterAZ	String	AZ of master nodes in a cluster. For details about AZs supported by CCE, see Regions and Endpoints . <ul style="list-style-type: none"> multi_az: (Optional) The cluster will span across AZs. Multiple AZs can be configured only when a cluster with multiple master nodes is used. <i>AZ of the dedicated cloud computing pool</i>: The cluster will be deployed in the DeC AZ. It is mandatory for DeC CCE clusters.

Parameter	Type	Description
dssMasterVolumes	String	<p>Whether the system and data disks of a master node use dedicated distributed storage. If this parameter is omitted or left unspecified, EVS disks are used by default.</p> <p>This parameter is mandatory for dedicated CCE clusters. It is in the following format: <code><rootVol.dssPoolID>.<rootVol.volType>;<dataVol.dssPoolID>.<dataVol.volType></code></p> <p>Field description:</p> <ul style="list-style-type: none"> • rootVol is the system disk. dataVol is the data disk. • dssPoolID indicates the ID of the DSS storage pool. • volType indicates the storage volume type of the DSS storage pool, such as SAS and SSD. <p>Example: <code>c950ee97-587c-4f24-8a74-3367e3da570f.sas;6edbc2f4-1507-44f8-ac0d-eed1d2608d38.ssd</code></p> <p>NOTE This field cannot be configured for non-dedicated CCE clusters.</p>
enterpriseProjectId	String	<p>ID of the enterprise project that a cluster belongs to.</p> <p>NOTE</p> <ul style="list-style-type: none"> • An enterprise project can be configured only after the enterprise project function is enabled. • The enterprise project to which the cluster belongs must be the same as that to which other cloud service resources associated with the cluster belong.

Parameter	Type	Description
kubeProxyMode	String	<p>Service forwarding mode. Two modes are available:</p> <ul style="list-style-type: none"> • iptables: Traditional kube-proxy uses iptables rules to implement Service load balancing. In this mode, too many iptables rules will be generated when many Services are deployed. In addition, non-incremental updates will cause latency and even tangible performance issues in the case of service traffic spikes. • ipvs: Optimized kube-proxy mode with higher throughput and faster speed. This mode supports incremental updates and can keep connections uninterrupted during Service updates. It is suitable for large-sized clusters. <p>NOTE This parameter has been deprecated. If this parameter and kubeProxyMode in ClusterSpec are specified at the same time, the latter is used.</p>
clusterExternalIP	String	EIP of the master node
alpha.cce/fixPoolMask	String	<p>Number of mask bits of the fixed IP address pool of the container network model. This field is supported only for the VPC network model (vpc-router).</p> <p>This parameter determines the number of container IP addresses that can be allocated to a node. The maximum number of pods that can be created on a node is decided by this parameter and maxPods set during node creation.</p> <p>For details, see Maximum Number of Pods That Can Be Created on a Node.</p> <p>For integer characters, the value ranges from 24 to 28.</p>
decMasterFlavor	String	Specifications of the master node in the dedicated hybrid cluster.
dockerUmaskMode	String	Default UmaskMode configuration of Docker in a cluster. The value can be secure or normal . If this parameter is not specified, normal is used by default.

Parameter	Type	Description
kubernetes.io/cpuManagerPolicy	String	<p>Cluster CPU management policy. The value can be none (or null) or static. The default value is none (or null).</p> <ul style="list-style-type: none"> • none or null: disables pods from exclusively occupying CPUs. Select this option if you want a large pool of shareable CPU cores. • static: enables pods to exclusively occupy CPUs. Select this option if your workload is sensitive to CPU cache and scheduling latency. In a CCE Turbo cluster, this setting is valid only for nodes where common containers, not Kata containers, run.
orderId	String	<p>Order ID. This parameter is returned in the response when the cluster is billed on a yearly/monthly basis with auto payment enabled (only in creation scenarios).</p>
periodType	String	<ul style="list-style-type: none"> • month: The unit is month. • year: The unit is year. <p>NOTE Request parameter, which is valid and mandatory when billingMode is set to 1 (yearly/monthly billing). Response parameter, which is returned only when a yearly/monthly cluster is created.</p>
periodNum	Integer	<p>Subscription duration. The value can be:</p> <ul style="list-style-type: none"> • If periodType is month, the value ranges from 1 to 9. • If periodType is year, the value ranges from 1 to 3. <p>NOTE Request parameter, which is valid and mandatory when billingMode is set to 1. Response parameter, which is returned only when a yearly/monthly cluster is created.</p>
isAutoRenew	String	<p>Whether auto renewal is enabled.</p> <ul style="list-style-type: none"> • true: Auto renewal is enabled. • false: Auto renewal is not enabled. <p>NOTE This field is valid when billingMode is set to 1. If not specified, auto renewal is not enabled.</p>

Parameter	Type	Description
isAutoPay	String	Whether to deduct fees automatically. <ul style="list-style-type: none"> • true: Enable automatic fee deduction. • false: Do not enable automatic fee deduction. <p>NOTE This field is valid when billingMode is set to 1. If not specified, auto fee deduction is not enabled.</p>
upgradefrom	String	Records of how the cluster is upgraded to the current version

Table 4-79 PackageConfiguration

Parameter	Type	Description
name	String	Component name.
configurations	Array of ConfigurationItem objects	Component configuration item.

Table 4-80 ConfigurationItem

Parameter	Type	Description
name	String	Component configuration item name.
value	Object	Component configuration item value.

Table 4-81 ClusterStatus

Parameter	Type	Description
phase	String	<p>Cluster status. Options:</p> <ul style="list-style-type: none"> ● Available: The cluster is running properly. ● Unavailable: The cluster is exhibiting unexpected behavior. Manually delete it. ● ScalingUp: Nodes are being added to the cluster. ● ScalingDown: The cluster is being downsized to fewer nodes. ● Creating: The cluster is being created. ● Deleting: The cluster is being deleted. ● Upgrading: The cluster is being upgraded. ● Resizing: Cluster specifications are being changed. ● RollingBack: The cluster is being rolled back. ● RollbackFailed: The cluster rollback is abnormal. ● Hibernating: The cluster is being hibernated. ● Hibernation: The cluster is in hibernation. ● Awaking: The cluster is being woken up from hibernation. ● Empty: The cluster does not have any resources. This field is discarded. ● Error: Resources in the cluster are abnormal. Manually delete the cluster.
jobID	String	<p>ID of the task associated with the cluster in the current state. Options:</p> <ul style="list-style-type: none"> ● ID of the associated task returned when creating a cluster. You can use it to obtain the auxiliary tasks for creating a cluster. ● ID of the associated task returned when a cluster fails to be deleted or is deleted. If this parameter is not empty, you can use the task ID to obtain the auxiliary tasks for deleting a cluster. <p>NOTE Tasks are short-lived. Do not use task information in scenarios such as cluster status determination.</p>
reason	String	Reason of cluster state change. This parameter is returned if the cluster is not in the Available state.

Parameter	Type	Description
message	String	Detailed information about why the cluster changes to the current state. This parameter is returned if the cluster is not in the Available state.
endpoints	Array of ClusterEndpoints objects	Access address of kube-apiserver in the cluster.
isLocked	Boolean	The CBC resource is locked.
lockScene	String	Scenario where the CBC resource is locked.
lockSource	String	Resource locking.
lockSourceId	String	ID of the locked resource.
deleteOption	Object	Whether to delete configurations. This parameter is contained only in the response to the deletion request.
deleteStatus	Object	Whether to delete the status information. This parameter is contained only in the response to the deletion request.

Table 4-82 ClusterEndpoints

Parameter	Type	Description
url	String	Access address of kube-apiserver in the cluster.
type	String	Type of the cluster access address. <ul style="list-style-type: none"> • Internal: address for internal network access • External: address for external network access

Example Requests

None

Example Responses

Status code: 200

The cluster list is successfully obtained.

```
{
  "kind": "Cluster",
  "apiVersion": "v3",
  "items": [ {
```

```
"kind" : "Cluster",
"apiVersion" : "v3",
"metadata" : {
  "alias" : "mycluster",
  "name" : "mycluster",
  "uid" : "4d1ecb2c-229a-11e8-9c75-0255ac100ceb",
  "creationTimestamp" : "2018-08-02 03:48:58.968214406 +0000 UTC",
  "updateTimestamp" : "2018-08-02 04:05:29.386391813 +0000 UTC"
},
"spec" : {
  "type" : "VirtualMachine",
  "flavor" : "cce.s1.small",
  "version" : "v1.7.3-r13",
  "description" : "awesome cluster",
  "customSan" : [ "192.168.1.0", "example.com" ],
  "hostNetwork" : {
    "vpc" : "f0c12911-4fdb-4284-9230-7ffb0860826a",
    "subnet" : "ac274229-fd2e-4695-9f01-a0c1372b8006"
  },
  "containerNetwork" : {
    "mode" : "overlay_l2",
    "cidr" : "172.16.0.0/16"
  },
  "authentication" : {
    "mode" : "x509",
    "authenticatingProxy" : { }
  },
  "billingMode" : 0
},
"status" : {
  "phase" : "Available",
  "endpoints" : [ {
    "url" : "https://192.168.0.11:5443",
    "type" : "Internal"
  } ]
}
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ListClustersSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";
```



```
ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

CceClient client = CceClient.newBuilder()
    .withCredential(auth)
    .withRegion(CceRegion.valueOf("<YOUR REGION>"))
    .build();
ListClustersRequest request = new ListClustersRequest();
try {
    ListClustersResponse response = client.listClusters(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListClustersRequest()
        response = client.list_clusters(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
```

```

"fmt"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListClustersRequest{}
    response, err := client.ListClusters(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The cluster list is successfully obtained.

Error Codes

See [Error Codes](#).

4.2.4 Updating a Specified Cluster

Function

This API is used to update information about a specified cluster.

 NOTE

The URL for cluster management is in the format of https://Endpoint/uri. In the URL, **uri** indicates the resource path, that is, the path for API access.

Calling Method

For details, see [Calling APIs](#).

URI

PUT /api/v3/projects/{project_id}/clusters/{cluster_id}

Table 4-83 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-84 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Table 4-85 Request body parameters

Parameter	Mandatory	Type	Description
spec	Yes	ClusterInformationSpec object	Detailed cluster parameters
metadata	No	ClusterMetadataForUpdate object	Basic cluster information, including name-related fields

Table 4-86 ClusterInformationSpec

Parameter	Mandatory	Type	Description
description	No	String	Cluster description. <ol style="list-style-type: none"> 1. A maximum of 200 characters are allowed. The value cannot contain the following special characters: ~\$%^&*<>[]{}()'"#\ 2. Only clusters in the Available, ScalingUp, and ScalingDown states can be modified.
customSan	No	Array of strings	Custom SAN field in the server certificate of the cluster API server, which must comply with the SSL and X509 format specifications. <ol style="list-style-type: none"> 1. Duplicate names are not allowed. 2. Must comply with the IP address and domain name formats. <p>Example:</p> <pre>SAN 1: DNS Name=example.com SAN 2: DNS Name=www.example.com SAN 3: DNS Name=example.net SAN 4: IP Address=93.184.216.34</pre>
containerNetwork	No	ContainerNetworkUpdate object	Container networking parameters, including information about the container CIDR block.

Parameter	Mandatory	Type	Description
eniNetwork	No	EniNetworkUpdate object	Cloud Native Network 2.0 network configuration, including the container subnet information of the CCE Turbo cluster.
hostNetwork	No	hostNetwork object	Node network parameters, including the default security group settings.

Table 4-87 ContainerNetworkUpdate

Parameter	Mandatory	Type	Description
cidrs	No	Array of ContainerCIDR objects	List of container CIDR blocks. For a clusters of v1.21 or later, if it uses a VPC network, a maximum of 20 container CIDR blocks can be added incrementally. The configuration cannot be modified after the cluster is updated.

Table 4-88 ContainerCIDR

Parameter	Mandatory	Type	Description
cidr	Yes	String	Container CIDR block. Recommended: 10.0.0.0/12-19, 172.16.0.0/16-19, and 192.168.0.0/16-19

Table 4-89 EniNetworkUpdate

Parameter	Mandatory	Type	Description
subnets	No	Array of NetworkSubnet objects	<p>List of IPv4 subnet IDs. CCE Turbo clusters of version 1.19.10 and later support multiple container subnets. In addition, the container subnet list can be incrementally updated.</p> <p>Subnets can be added, not deleted. Exercise caution when selecting a subnet.</p> <p>The request body must contain all existing subnets.</p>

Table 4-90 NetworkSubnet

Parameter	Mandatory	Type	Description
subnetID	Yes	String	<p>IPv4 subnet ID of the subnet for creating master nodes. IPv6 is not supported. You can obtain it in either of the following ways:</p> <ul style="list-style-type: none"> • Method 1: Log in to the VPC console and click the target subnet on the Subnets page. You can view the IPv4 subnet ID on the displayed page. • Method 2: Use the VPC API to obtain subnets. <p>For details, see Querying Subnets.</p>

Table 4-91 hostNetwork

Parameter	Mandatory	Type	Description
SecurityGroup	No	String	<p>The default security group of the worker node in the cluster needs to allow traffic through some ports to ensure normal communication. For details, see How Do I Harden the Automatically Created Security Group Rules for CCE Cluster Nodes.</p> <p>The modified security group applies only to nodes newly created or accepted. For existing nodes, you need to manually modify the security group rules for them.</p>

Table 4-92 ClusterMetadataForUpdate

Parameter	Mandatory	Type	Description
alias	No	String	<p>Alias of a cluster name.</p> <p>Enter 4 to 128 characters, starting with a lowercase letter and not ending with a hyphen (-). Only lowercase letters, digits, and hyphens (-) are allowed.</p> <p>A cluster alias must be unique.</p> <p>If this parameter is left blank, the alias is not changed.</p>

Response Parameters

Status code: 200

Table 4-93 Response body parameters

Parameter	Type	Description
kind	String	API type. The value is fixed at Cluster or cluster and cannot be changed.
apiVersion	String	API version. The value is fixed at v3 and cannot be changed.

Parameter	Type	Description
metadata	ClusterMetadata object	Basic information about a cluster. Metadata is a collection of attributes.
spec	ClusterSpec object	Detailed description of the cluster. CCE creates or updates objects by defining or updating spec .
status	ClusterStatus object	Cluster status and job ID of the cluster creation job.

Table 4-94 ClusterMetadata

Parameter	Type	Description
name	String	Cluster name. Enter 4 to 128 characters, starting with a lowercase letter and not ending with a hyphen (-). Only lowercase letters, digits, and hyphens (-) are allowed.
uid	String	Cluster ID, which uniquely identifies a cluster. This ID is automatically generated after a cluster is created. Only the automatically generated ID will take effect. When you create a yearly/monthly cluster, no cluster ID will be returned in the response body.
alias	String	Alias of a cluster name displayed on the CCE console, and the name can be changed. Enter 4 to 128 characters, starting with a lowercase letter and not ending with a hyphen (-). Only lowercase letters, digits, and hyphens (-) are allowed. A cluster alias must be unique. In the request body for creating or updating a cluster, if the cluster alias is not specified or set to null, the cluster name will be used as the cluster alias. In the response body for obtaining a cluster, the cluster alias is returned. If it is not configured, the cluster name will be returned.

Parameter	Type	Description
annotations	Map<String,String>	<p>Cluster annotations, in the format of key-value pairs.</p> <pre>"annotations": { "key1" : "value1", "key2" : "value2" }</pre> <p>NOTE</p> <ul style="list-style-type: none"> • annotations: Does not label or select objects. The metadata in annotations may be small or large, structured or unstructured, and may include characters that are not allowed in labels. • This field is not stored in the database and is used only to specify the add-ons to be installed in the cluster. • Install ICAgent during cluster creation by adding the key-value pair "cluster.install.addons.external/install":{"addonTemplateName":"icagent"}".
labels	Map<String,String>	<p>Cluster labels, in the format of key-value pairs.</p> <p>NOTE</p> <p>The value of this field is automatically generated by the system and is used by the frontend to identify the features supported by the cluster during the upgrade. Customized values are invalid.</p>
creationTimestamp	String	Time when the cluster was created.
updateTimestamp	String	Time when the cluster was updated.

Table 4-95 ClusterSpec

Parameter	Type	Description
category	String	<p>Cluster type. Options:</p> <ul style="list-style-type: none"> • CCE: CCE cluster CCE cluster supports hybrid deployment of VMs and BMSs, and heterogeneous nodes such as GPU and NPU nodes, allowing you to run your containers in a secure and stable container runtime environment based on a high-performance network model. • Turbo: CCE Turbo cluster One-stop, cost-effective CCE Turbo clusters run on the cloud native 2.0 infrastructure featuring hardware-software synergy for lossless networking, high security and reliability, and intelligent scheduling.

Parameter	Type	Description
type	String	Master node architecture: <ul style="list-style-type: none"> ● VirtualMachine: x86 ● ARM64: Arm-based Kunpeng
flavor	String	Cluster specifications. Specifications of clusters 1.15 or later versions can be changed after they are created. For details, see Modifying Cluster Specifications . Options: <ul style="list-style-type: none"> ● cce.s1.small: a small-scale CCE cluster with one master node and a maximum of 50 worker nodes ● cce.s1.medium: a medium-scale CCE cluster with one master node and a maximum of 200 worker nodes ● cce.s2.small: a small-scale CCE cluster with three master nodes and a maximum of 50 worker nodes ● cce.s2.medium: a medium-scale CCE cluster with three master nodes and a maximum of 200 worker nodes ● cce.s2.large: a large-scale CCE cluster with three master nodes and a maximum of 1,000 worker nodes ● cce.s2.xlarge: an ultra-large-scale CCE cluster with three master nodes and a maximum of 2,000 worker nodes <p>NOTE The fields in the parameters are described as follows:</p> <ul style="list-style-type: none"> ● s1: specifies a cluster with one master node. If the master node is faulty, the cluster will become unavailable, but running workloads in the cluster are not affected. ● s2: specifies an HA cluster with three master nodes. If one of the master nodes is faulty, the cluster is still available. ● dec: specifies a DeC CCE cluster. For example, cce.dec.s1.small specifies a small-scale, DeC CCE cluster with one master node and a maximum of 50 worker nodes. ● small: specifies that a cluster can manage a maximum of 50 worker nodes. ● medium: specifies that a cluster can manage a maximum of 200 worker nodes. ● large: specifies that a cluster can manage a maximum of 1,000 worker nodes. ● xlarge: specifies that a cluster can manage a maximum of 2,000 worker nodes.

Parameter	Type	Description
version	String	<p>Cluster version, which mirrors the baseline version of the Kubernetes community. The latest version is recommended.</p> <p>You can create clusters of two latest versions on the CCE console. To learn which cluster versions are available, log in to the CCE console, create a cluster, and check the Cluster Version parameter.</p> <p>You can call APIs to create clusters of other versions. However, these cluster versions will be gradually terminated. For details about the support policy, see the CCE announcement.</p> <p>NOTE</p> <ul style="list-style-type: none"> • If not specified, a cluster of the latest version will be created. • If a baseline cluster version is specified but the R version is not specified, a cluster of the latest R version will be created by default. It is a good practice not to specify the R version. • CCE Turbo clusters of v1.19 or later are commercially available.
platformVersion	String	<p>CCE cluster platform version, indicating the internal version under the cluster version (version). Platform versions are used to trace iterations in a major cluster version. They are unique within a major cluster version and recounted when the major cluster version changes. This parameter cannot be customized. When you create a cluster, the latest corresponding platform version is automatically selected.</p> <p>The format of platformVersion is cce.X.Y.</p> <ul style="list-style-type: none"> • X: internal feature version, indicating changes in features, patches, or OS support in the cluster version. The value starts from 1 and increases monotonically. • Y: patch version of an internal feature version. It is used only for software package update after the feature version is brought online. No other modification is involved. The value starts from 0 and increases monotonically.

Parameter	Type	Description
description	String	Cluster description, for example, which purpose the cluster is intended to serve. By default, this field is left unspecified. To modify cluster description after the cluster is created, call the API for updating cluster information or go to the cluster details page on the CCE console. Only UTF-8 encoding is supported.
customSan	Array of strings	Custom SAN field in the server certificate of the cluster API server, which must comply with the SSL and X509 format specifications. <ol style="list-style-type: none"> 1. Duplicate names are not allowed. 2. Must comply with the IP address and domain name formats. <p>Example:</p> <pre>SAN 1: DNS Name=example.com SAN 2: DNS Name=www.example.com SAN 3: DNS Name=example.net SAN 4: IP Address=93.184.216.34</pre>
ipv6enable	Boolean	Whether the cluster supports IPv6 addresses. This field is supported in clusters of v1.15 and later versions.
hostNetwork	HostNetwork object	Node networking parameters, including VPC and subnet ID. This field is mandatory because nodes in a cluster communicate with each other by using a VPC.
containerNetwork	ContainerNetwork object	Container networking parameters, including the container network model and container CIDR block.
eniNetwork	EniNetwork object	Configuration of the Cloud Native Network 2.0 model. Specify this field when creating a CCE Turbo cluster.
serviceNetwork	ServiceNetwork object	Service CIDR block, including IPv4 CIDR blocks.
authentication	Authentication object	Configurations of the cluster authentication mode.
billingMode	Integer	Billing mode of a cluster. <ul style="list-style-type: none"> • 0: pay-per-use • 1: yearly/monthly Defaults to pay-per-use.
masters	Array of MasterSpec objects	Advanced configurations of master nodes

Parameter	Type	Description
kubernetesSvcIpRange	String	Service CIDR blocks for Kubernetes clusterIPs. This field is available only for clusters of v1.11.7 and later. If this parameter is not specified during cluster creation, the default value 10.247.0.0/16 will be used. This parameter is deprecated. Use serviceNetwork instead. The new field contains the IPv4 CIDR blocks.
clusterTags	Array of ResourceTag objects	Cluster resource tags.
kubeProxyMode	String	Service forwarding mode. Options: <ul style="list-style-type: none"> • iptables: Traditional kube-proxy uses iptables rules to implement Service load balancing. In this mode, too many iptables rules will be generated when many Services are deployed. In addition, non-incremental updates will cause latency and even tangible performance issues in the case of service traffic spikes. • ipvs: Optimized kube-proxy mode with higher throughput and faster speed. This mode supports incremental updates and can keep connections uninterrupted during Service updates. It is suitable for large-sized clusters. <p>NOTE iptables is used by default.</p>
az	String	AZ. This field is returned only for a query. For details about AZs supported by CCE, see Regions and Endpoints .
extendParam	ClusterExtendedParam object	Extended field to decide whether the cluster will span across AZs or belong to a specified enterprise project, or whether a dedicated CCE cluster is to be created.
supportIstio	Boolean	Whether Istio is supported.
configurationsOverride	Array of PackageConfiguration objects	Cluster default component configuration override. If you specify a component or parameter that is not supported, the configuration item will be ignored. For details about the supported components and their parameters, see Cluster Configuration Management .

Table 4-96 HostNetwork

Parameter	Type	Description
vpc	String	<p>ID of the VPC used to create a master node. You can obtain it in either of the following ways:</p> <ul style="list-style-type: none"> • Method 1: Log in to the VPC console and view the VPC ID in the VPC details page. • Method 2: Use the VPC API to obtain VPC IDs. <p>For details, see Querying VPCs.</p>
subnet	String	<p>Network ID of the subnet used to create a master node. You can obtain it in either of the following ways:</p> <ul style="list-style-type: none"> • Method 1: Log in to the VPC console and click the target subnet on the Subnets page. You can view the network ID on the displayed page. • Method 2: Use the VPC API to obtain subnets. <p>For details, see Querying Subnets.</p>
SecurityGroup	String	<p>Default worker node security group ID of the cluster. If specified, the cluster will be bound to the target security group. Otherwise, the system will automatically create a default worker node security group for you. The default worker node security group needs to allow access from certain ports to ensure normal communications. For details, see How Do I Harden the Automatically Created Security Group Rules for CCE Cluster Nodes.</p>

Table 4-97 ContainerNetwork

Parameter	Type	Description
mode	String	<p>Container network model. Select one of the following possible values:</p> <ul style="list-style-type: none"> • overlay_l2: an overlay_l2 network (container tunnel network) built for containers by using OpenVSwitch (OVS). • vpc-router: an underlay_l2 network built for containers by using IPvlan and custom VPC routes. • eni: Cloud Native Network 2.0. This model has integrated cloud native elastic network interfaces (ENIs), uses VPC CIDR blocks to allocate container IP addresses, and allows direct traffic distribution to containers through a load balancer for high performance. Use this model when creating a CCE Turbo cluster.
cidr	String	<p>Container CIDR block. Recommended: 10.0.0.0/12-19, 172.16.0.0/16-19, or 192.168.0.0/16-19. If the selected CIDR block conflicts with existing ones, an error will be reported.</p> <p>Not editable after the cluster is created. (deprecated. A specified cidrs will make cidr invalid.)</p>
cidrs	Array of ContainerCIDR objects	<p>List of container CIDR blocks. In clusters of v1.21 or later, the cidrs field is used. When the cluster network type is vpc-router, you can configure a maximum of 20 container CIDR blocks. In cluster versions earlier than v1.21, if the cidrs field is used, the first CIDR element in the array is used as the container CIDR block.</p> <p>The configuration cannot be modified after the cluster is created.</p>

Table 4-98 ContainerCIDR

Parameter	Type	Description
cidr	String	<p>Container CIDR block. Recommended: 10.0.0.0/12-19, 172.16.0.0/16-19, and 192.168.0.0/16-19</p>

Table 4-99 EniNetwork

Parameter	Type	Description
eniSubnetId	String	IPv4 subnet ID of ENI subnet. (IPv6 is not supported and is being discarded.) You can obtain it in either of the following ways: <ul style="list-style-type: none"> Method 1: Log in to the VPC console and click the target subnet on the Subnets page. You can view the IPv4 subnet ID on the displayed page. Method 2: Use the VPC API to obtain subnet IDs. For details, see Querying Subnets .
eniSubnetCIDR	String	ENI subnet CIDR (being discarded)
subnets	Array of NetworkSubnet objects	List of IPv4 subnet IDs

Table 4-100 NetworkSubnet

Parameter	Type	Description
subnetID	String	IPv4 subnet ID of the subnet for creating master nodes. IPv6 is not supported. You can obtain it in either of the following ways: <ul style="list-style-type: none"> Method 1: Log in to the VPC console and click the target subnet on the Subnets page. You can view the IPv4 subnet ID on the displayed page. Method 2: Use the VPC API to obtain subnets. For details, see Querying Subnets .

Table 4-101 ServiceNetwork

Parameter	Type	Description
IPv4CIDR	String	Value range of the Kubernetes clusterIP IPv4 CIDR blocks. If this parameter is not specified during cluster creation, the default value 10.247.0.0/16 will be used.

Table 4-102 Authentication

Parameter	Type	Description
mode	String	Cluster authentication mode. <ul style="list-style-type: none"> Clusters of Kubernetes v1.11 or earlier support x509, rbac, and authenticating_proxy. Defaults to x509. Clusters of Kubernetes v1.13 or later support rbac and authenticating_proxy. Defaults to rbac.
authenticatingProxy	AuthenticatingProxy object	Configuration related to the <code>authenticating_proxy</code> mode. This field is mandatory when the authentication mode is authenticating_proxy .

Table 4-103 AuthenticatingProxy

Parameter	Type	Description
ca	String	X509 CA certificate (Base64-encoded) configured in <code>authenticating_proxy</code> mode. This field is mandatory when the cluster authentication mode is authenticating_proxy . Maximum size: 1 MB
cert	String	Client certificate issued by the X509 CA certificate configured in <code>authenticating_proxy</code> mode, which is used for authentication from kube-apiserver to the extended API server. (The value must be Base64-encoded.) This field is mandatory when the cluster authentication mode is authenticating_proxy .
privateKey	String	Private key of the client certificate issued by the X509 CA certificate configured in <code>authenticating_proxy</code> mode, which is used for authentication from kube-apiserver to the extended API server. The private key used by the Kubernetes cluster does not support password encryption. Use an unencrypted private key. (The value must be Base64-encoded.) This field is mandatory when the cluster authentication mode is authenticating_proxy .

Table 4-104 MasterSpec

Parameter	Type	Description
availabilityZone	String	AZ

Table 4-105 ResourceTag

Parameter	Type	Description
key	String	Key. <ul style="list-style-type: none"> • Cannot be null. Max characters: 128. • Use letters, digits, and spaces in UTF-8 format. • Can contain the following special characters: <code>._:/=+-@</code>. • Cannot start with <code>_sys_</code>.
value	String	Value. <ul style="list-style-type: none"> • Can be null but not the default. Max characters: 255. • Use letters, digits, and spaces in UTF-8 format. • Can contain the following special characters: <code>._:/=+-@</code>.

Table 4-106 ClusterExtendParam

Parameter	Type	Description
clusterAZ	String	AZ of master nodes in a cluster. For details about AZs supported by CCE, see Regions and Endpoints . <ul style="list-style-type: none"> • multi_az: (Optional) The cluster will span across AZs. Multiple AZs can be configured only when a cluster with multiple master nodes is used. • <i>AZ of the dedicated cloud computing pool</i>: The cluster will be deployed in the DeC AZ. It is mandatory for DeC CCE clusters.

Parameter	Type	Description
dssMasterVolumes	String	<p>Whether the system and data disks of a master node use dedicated distributed storage. If this parameter is omitted or left unspecified, EVS disks are used by default.</p> <p>This parameter is mandatory for dedicated CCE clusters. It is in the following format: <code><rootVol.dssPoolID>.<rootVol.volType>;<dataVol.dssPoolID>.<dataVol.volType></code></p> <p>Field description:</p> <ul style="list-style-type: none"> • rootVol is the system disk. dataVol is the data disk. • dssPoolID indicates the ID of the DSS storage pool. • volType indicates the storage volume type of the DSS storage pool, such as SAS and SSD. <p>Example: <code>c950ee97-587c-4f24-8a74-3367e3da570f.sas;6edbc2f4-1507-44f8-ac0d-eed1d2608d38.ssd</code></p> <p>NOTE This field cannot be configured for non-dedicated CCE clusters.</p>
enterpriseProjectId	String	<p>ID of the enterprise project that a cluster belongs to.</p> <p>NOTE</p> <ul style="list-style-type: none"> • An enterprise project can be configured only after the enterprise project function is enabled. • The enterprise project to which the cluster belongs must be the same as that to which other cloud service resources associated with the cluster belong.

Parameter	Type	Description
kubeProxyMode	String	<p>Service forwarding mode. Two modes are available:</p> <ul style="list-style-type: none"> • iptables: Traditional kube-proxy uses iptables rules to implement Service load balancing. In this mode, too many iptables rules will be generated when many Services are deployed. In addition, non-incremental updates will cause latency and even tangible performance issues in the case of service traffic spikes. • ipvs: Optimized kube-proxy mode with higher throughput and faster speed. This mode supports incremental updates and can keep connections uninterrupted during Service updates. It is suitable for large-sized clusters. <p>NOTE This parameter has been deprecated. If this parameter and kubeProxyMode in ClusterSpec are specified at the same time, the latter is used.</p>
clusterExternalIP	String	EIP of the master node
alpha.cce/fixPoolMask	String	<p>Number of mask bits of the fixed IP address pool of the container network model. This field is supported only for the VPC network model (vpc-router).</p> <p>This parameter determines the number of container IP addresses that can be allocated to a node. The maximum number of pods that can be created on a node is decided by this parameter and maxPods set during node creation.</p> <p>For details, see Maximum Number of Pods That Can Be Created on a Node.</p> <p>For integer characters, the value ranges from 24 to 28.</p>
decMasterFlavor	String	Specifications of the master node in the dedicated hybrid cluster.
dockerUmaskMode	String	Default UmaskMode configuration of Docker in a cluster. The value can be secure or normal . If this parameter is not specified, normal is used by default.

Parameter	Type	Description
kubernetes.io/cpuManagerPolicy	String	<p>Cluster CPU management policy. The value can be none (or null) or static. The default value is none (or null).</p> <ul style="list-style-type: none"> ● none or null: disables pods from exclusively occupying CPUs. Select this option if you want a large pool of shareable CPU cores. ● static: enables pods to exclusively occupy CPUs. Select this option if your workload is sensitive to CPU cache and scheduling latency. In a CCE Turbo cluster, this setting is valid only for nodes where common containers, not Kata containers, run.
orderId	String	<p>Order ID. This parameter is returned in the response when the cluster is billed on a yearly/monthly basis with auto payment enabled (only in creation scenarios).</p>
periodType	String	<ul style="list-style-type: none"> ● month: The unit is month. ● year: The unit is year. <p>NOTE Request parameter, which is valid and mandatory when billingMode is set to 1 (yearly/monthly billing). Response parameter, which is returned only when a yearly/monthly cluster is created.</p>
periodNum	Integer	<p>Subscription duration. The value can be:</p> <ul style="list-style-type: none"> ● If periodType is month, the value ranges from 1 to 9. ● If periodType is year, the value ranges from 1 to 3. <p>NOTE Request parameter, which is valid and mandatory when billingMode is set to 1. Response parameter, which is returned only when a yearly/monthly cluster is created.</p>
isAutoRenew	String	<p>Whether auto renewal is enabled.</p> <ul style="list-style-type: none"> ● true: Auto renewal is enabled. ● false: Auto renewal is not enabled. <p>NOTE This field is valid when billingMode is set to 1. If not specified, auto renewal is not enabled.</p>

Parameter	Type	Description
isAutoPay	String	Whether to deduct fees automatically. <ul style="list-style-type: none"> • true: Enable automatic fee deduction. • false: Do not enable automatic fee deduction. <p>NOTE This field is valid when billingMode is set to 1. If not specified, auto fee deduction is not enabled.</p>
upgradefrom	String	Records of how the cluster is upgraded to the current version

Table 4-107 PackageConfiguration

Parameter	Type	Description
name	String	Component name.
configurations	Array of ConfigurationItem objects	Component configuration item.

Table 4-108 ConfigurationItem

Parameter	Type	Description
name	String	Component configuration item name.
value	Object	Component configuration item value.

Table 4-109 ClusterStatus

Parameter	Type	Description
phase	String	<p>Cluster status. Options:</p> <ul style="list-style-type: none"> ● Available: The cluster is running properly. ● Unavailable: The cluster is exhibiting unexpected behavior. Manually delete it. ● ScalingUp: Nodes are being added to the cluster. ● ScalingDown: The cluster is being downsized to fewer nodes. ● Creating: The cluster is being created. ● Deleting: The cluster is being deleted. ● Upgrading: The cluster is being upgraded. ● Resizing: Cluster specifications are being changed. ● RollingBack: The cluster is being rolled back. ● RollbackFailed: The cluster rollback is abnormal. ● Hibernating: The cluster is being hibernated. ● Hibernation: The cluster is in hibernation. ● Awaking: The cluster is being woken up from hibernation. ● Empty: The cluster does not have any resources. This field is discarded. ● Error: Resources in the cluster are abnormal. Manually delete the cluster.
jobID	String	<p>ID of the task associated with the cluster in the current state. Options:</p> <ul style="list-style-type: none"> ● ID of the associated task returned when creating a cluster. You can use it to obtain the auxiliary tasks for creating a cluster. ● ID of the associated task returned when a cluster fails to be deleted or is deleted. If this parameter is not empty, you can use the task ID to obtain the auxiliary tasks for deleting a cluster. <p>NOTE Tasks are short-lived. Do not use task information in scenarios such as cluster status determination.</p>
reason	String	Reason of cluster state change. This parameter is returned if the cluster is not in the Available state.

Parameter	Type	Description
message	String	Detailed information about why the cluster changes to the current state. This parameter is returned if the cluster is not in the Available state.
endpoints	Array of ClusterEndpoints objects	Access address of kube-apiserver in the cluster.
isLocked	Boolean	The CBC resource is locked.
lockScene	String	Scenario where the CBC resource is locked.
lockSource	String	Resource locking.
lockSourceId	String	ID of the locked resource.
deleteOption	Object	Whether to delete configurations. This parameter is contained only in the response to the deletion request.
deleteStatus	Object	Whether to delete the status information. This parameter is contained only in the response to the deletion request.

Table 4-110 ClusterEndpoints

Parameter	Type	Description
url	String	Access address of kube-apiserver in the cluster.
type	String	Type of the cluster access address. <ul style="list-style-type: none"> • Internal: address for internal network access • External: address for external network access

Example Requests

- Updating only the description of a cluster

```
{
  "spec": {
    "description": "new description"
  }
}
```

- Updating only the custom SAN of the cluster certificate

```
{
  "spec": {
    "customSan": [ "192.168.1.0", "example.com" ]
  }
}
```


- Updating both the cluster description and custom certificate SAN

```
{
  "spec" : {
    "description" : "new description",
    "customSan" : [ "192.168.1.0", "example.com" ]
  }
}
```

- Configure a CIDR block for a cluster that uses VPCs and is of a version later than v1.21.

```
{
  "spec" : {
    "containerNetwork" : {
      "cidrs" : [ {
        "cidr" : "10.10.0.0/16"
      }, {
        "cidr" : "10.11.0.0/16"
      } ]
    }
  }
}
```

- Modify the default security group of a node in a cluster.

/api/v3/projects/{project_id}/clusters/{cluster_id}

```
{
  "spec" : {
    "hostNetwork" : {
      "SecurityGroup" : "6ee29825-8f49-4796-b33a-fc76f84a59ae"
    }
  }
}
```

Example Responses

Status code: 200

Information about the specified cluster is successfully updated.

```
{
  "kind" : "Cluster",
  "apiVersion" : "v3",
  "metadata" : {
    "alias" : "mycluster",
    "name" : "mycluster",
    "uid" : "4d1ecb2c-229a-11e8-9c75-0255ac100ceb",
    "creationTimestamp" : "2018-08-02 03:48:58.968214406 +0000 UTC",
    "updateTimestamp" : "2018-08-02 06:39:36.844676088 +0000 UTC"
  },
  "spec" : {
    "type" : "VirtualMachine",
    "flavor" : "cce.s1.small",
    "version" : "v1.7.3-r13",
    "description" : "new description",
    "customSan" : [ "192.168.1.0", "example.com" ],
    "hostNetwork" : {
      "vpc" : "4d1ecb2c-229a-11e8-9c75-0255ac100ceb",
      "subnet" : "4d1ecb2c-229a-11e8-9c75-0255ac100ceb",
      "SecurityGroup" : "6ee29825-8f49-4796-b33a-fc76f84a59ae"
    },
    "containerNetwork" : {
      "mode" : "overlay_l2",
      "cidr" : "172.17.0.0/16"
    },
    "authentication" : {
      "mode" : "x509",
      "authenticatingProxy" : { }
    }
  }
}
```

```
    },
    "billingMode" : 0
  },
  "status" : {
    "phase" : "Available",
    "endpoints" : [ {
      "url" : "https://192.168.0.11:5443",
      "type" : "Internal"
    } ]
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

- Updating only the description of a cluster

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class UpdateClusterSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateClusterRequest request = new UpdateClusterRequest();
        request.withClusterId("{cluster_id}");
        ClusterInformation body = new ClusterInformation();
        ClusterInformationSpec specbody = new ClusterInformationSpec();
        specbody.withDescription("new description");
        body.withSpec(specbody);
        request.withBody(body);
        try {
            UpdateClusterResponse response = client.updateCluster(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        }
    }
}
```

```
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

- Updating only the custom SAN of the cluster certificate

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateClusterSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateClusterRequest request = new UpdateClusterRequest();
        request.withClusterId("{cluster_id}");
        ClusterInformation body = new ClusterInformation();
        List<String> listSpecCustomSan = new ArrayList<>();
        listSpecCustomSan.add("192.168.1.0");
        listSpecCustomSan.add("example.com");
        ClusterInformationSpec specbody = new ClusterInformationSpec();
        specbody.withCustomSan(listSpecCustomSan);
        body.withSpec(specbody);
        request.withBody(body);
        try {
            UpdateClusterResponse response = client.updateCluster(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

```
}  
}  
}
```

- Updating both the cluster description and custom certificate SAN

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cce.v3.region.CceRegion;  
import com.huaweicloud.sdk.cce.v3.*;  
import com.huaweicloud.sdk.cce.v3.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class UpdateClusterSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before  
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local  
        // environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        CceClient client = CceClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))  
            .build();  
        UpdateClusterRequest request = new UpdateClusterRequest();  
        request.withClusterId("{cluster_id}");  
        ClusterInformation body = new ClusterInformation();  
        List<String> listSpecCustomSan = new ArrayList<>();  
        listSpecCustomSan.add("192.168.1.0");  
        listSpecCustomSan.add("example.com");  
        ClusterInformationSpec specbody = new ClusterInformationSpec();  
        specbody.withDescription("new description")  
            .withCustomSan(listSpecCustomSan);  
        body.withSpec(specbody);  
        request.withBody(body);  
        try {  
            UpdateClusterResponse response = client.updateCluster(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

- Configure a CIDR block for a cluster that uses VPCs and is of a version later than v1.21.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateClusterSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateClusterRequest request = new UpdateClusterRequest();
        request.withClusterId("{cluster_id}");
        ClusterInformation body = new ClusterInformation();
        List<ContainerCIDR> listContainerNetworkCidrs = new ArrayList<>();
        listContainerNetworkCidrs.add(
            new ContainerCIDR()
                .withCidr("10.10.0.0/16")
        );
        listContainerNetworkCidrs.add(
            new ContainerCIDR()
                .withCidr("10.11.0.0/16")
        );
        ContainerNetworkUpdate containerNetworkSpec = new ContainerNetworkUpdate();
        containerNetworkSpec.withCidrs(listContainerNetworkCidrs);
        ClusterInformationSpec specbody = new ClusterInformationSpec();
        specbody.withContainerNetwork(containerNetworkSpec);
        body.withSpec(specbody);
        request.withBody(body);
        try {
            UpdateClusterResponse response = client.updateCluster(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

```
    }  
  }  
}
```

- **Modify the default security group of a node in a cluster.**

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cce.v3.region.CceRegion;  
import com.huaweicloud.sdk.cce.v3.*;  
import com.huaweicloud.sdk.cce.v3.model.*;  
  
public class UpdateClusterSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before  
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local  
        // environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        CceClient client = CceClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))  
            .build();  
        UpdateClusterRequest request = new UpdateClusterRequest();  
        request.withClusterId("{cluster_id}");  
        ClusterInformation body = new ClusterInformation();  
        ClusterInformationSpecHostNetwork hostNetworkSpec = new  
ClusterInformationSpecHostNetwork();  
        hostNetworkSpec.withSecurityGroup("6ee29825-8f49-4796-b33a-fc76f84a59ae");  
        ClusterInformationSpec specbody = new ClusterInformationSpec();  
        specbody.withHostNetwork(hostNetworkSpec);  
        body.withSpec(specbody);  
        request.withBody(body);  
        try {  
            UpdateClusterResponse response = client.updateCluster(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

Python

- Updating only the description of a cluster

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateClusterRequest()
        request.cluster_id = "{cluster_id}"
        specbody = ClusterInformationSpec(
            description="new description"
        )
        request.body = ClusterInformation(
            spec=specbody
        )
        response = client.update_cluster(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Updating only the custom SAN of the cluster certificate

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
```

```
.build()

try:
    request = UpdateClusterRequest()
    request.cluster_id = "{cluster_id}"
    listCustomSanSpec = [
        "192.168.1.0",
        "example.com"
    ]
    specbody = ClusterInformationSpec(
        custom_san=listCustomSanSpec
    )
    request.body = ClusterInformation(
        spec=specbody
    )
    response = client.update_cluster(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

- Updating both the cluster description and custom certificate SAN

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *
```

```
if __name__ == "__main__":
```

```
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
```

```
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"
```

```
    credentials = BasicCredentials(ak, sk, projectId) \
```

```
    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()
```

```
try:
    request = UpdateClusterRequest()
    request.cluster_id = "{cluster_id}"
    listCustomSanSpec = [
        "192.168.1.0",
        "example.com"
    ]
    specbody = ClusterInformationSpec(
        description="new description",
        custom_san=listCustomSanSpec
    )
    request.body = ClusterInformation(
        spec=specbody
    )
    response = client.update_cluster(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```


- Configure a CIDR block for a cluster that uses VPCs and is of a version later than v1.21.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateClusterRequest()
        request.cluster_id = "{cluster_id}"
        listCidrsContainerNetwork = [
            ContainerCIDR(
                cidr="10.10.0.0/16"
            ),
            ContainerCIDR(
                cidr="10.11.0.0/16"
            )
        ]
        containerNetworkSpec = ContainerNetworkUpdate(
            cidrs=listCidrsContainerNetwork
        )
        specbody = ClusterInformationSpec(
            container_network=containerNetworkSpec
        )
        request.body = ClusterInformation(
            spec=specbody
        )
        response = client.update_cluster(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Modify the default security group of a node in a cluster.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
```

```
environment
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId) \

client = CceClient.new_builder() \
.with_credentials(credentials) \
.with_region(CceRegion.value_of("<YOUR REGION>")) \
.build()

try:
request = UpdateClusterRequest()
request.cluster_id = "{cluster_id}"
hostNetworkSpec = ClusterInformationSpecHostNetwork(
    security_group="6ee29825-8f49-4796-b33a-fc76f84a59ae"
)
specbody = ClusterInformationSpec(
    host_network=hostNetworkSpec
)
request.body = ClusterInformation(
    spec=specbody
)
response = client.update_cluster(request)
print(response)
except exceptions.ClientRequestException as e:
print(e.status_code)
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

Go

- Updating only the description of a cluster

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateClusterRequest{}
```

```

request.ClusterId = "{cluster_id}"
descriptionSpec:= "new description"
specbody := &model.ClusterInformationSpec{
    Description: &descriptionSpec,
}
request.Body = &model.ClusterInformation{
    Spec: specbody,
}
response, err := client.UpdateCluster(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}

```

- Updating only the custom SAN of the cluster certificate

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateClusterRequest{}
    request.ClusterId = "{cluster_id}"
    var listCustomSanSpec = []string{
        "192.168.1.0",
        "example.com",
    }
    specbody := &model.ClusterInformationSpec{
        CustomSan: &listCustomSanSpec,
    }
    request.Body = &model.ClusterInformation{
        Spec: specbody,
    }
    response, err := client.UpdateCluster(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

- Updating both the cluster description and custom certificate SAN

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateClusterRequest{}
    request.ClusterId = "{cluster_id}"
    var listCustomSanSpec = []string{
        "192.168.1.0",
        "example.com",
    }
    descriptionSpec := "new description"
    specbody := &model.ClusterInformationSpec{
        Description: &descriptionSpec,
        CustomSan: &listCustomSanSpec,
    }
    request.Body = &model.ClusterInformation{
        Spec: specbody,
    }
    response, err := client.UpdateCluster(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Configure a CIDR block for a cluster that uses VPCs and is of a version later than v1.21.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := cce.NewCceClient(
    cce.CceClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateClusterRequest{}
request.ClusterId = "{cluster_id}"
var listCidrsContainerNetwork = []model.ContainerCidr{
    {
        Cidr: "10.10.0.0/16",
    },
    {
        Cidr: "10.11.0.0/16",
    },
}
containerNetworkSpec := &model.ContainerNetworkUpdate{
    Cidrs: &listCidrsContainerNetwork,
}
specbody := &model.ClusterInformationSpec{
    ContainerNetwork: containerNetworkSpec,
}
request.Body = &model.ClusterInformation{
    Spec: specbody,
}
response, err := client.UpdateCluster(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

- Modify the default security group of a node in a cluster.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"
```

```

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := cce.NewCceClient(
    cce.CceClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateClusterRequest{}
request.ClusterId = "{cluster_id}"
securityGroupHostNetwork:= "6ee29825-8f49-4796-b33a-fc76f84a59ae"
hostNetworkSpec := &model.ClusterInformationSpecHostNetwork{
    SecurityGroup: &securityGroupHostNetwork,
}
specbody := &model.ClusterInformationSpec{
    HostNetwork: hostNetworkSpec,
}
request.Body = &model.ClusterInformation{
    Spec: specbody,
}
response, err := client.UpdateCluster(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Information about the specified cluster is successfully updated.

Error Codes

See [Error Codes](#).

4.2.5 Deleting a Cluster

Function

This API is used to delete a specified cluster.

NOTE

The URL for cluster management is in the format of https://Endpoint/uri. In the URL, **uri** indicates the resource path, that is, the path for API access.

Calling Method

For details, see [Calling APIs](#).

URI

DELETE /api/v3/projects/{project_id}/clusters/{cluster_id}

Table 4-111 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Table 4-112 Query Parameters

Parameter	Mandatory	Type	Description
delete_efs	No	String	Whether to delete SFS Turbo volumes. Value options: <ul style="list-style-type: none"> • true or block (The system starts to delete the object. If the deletion fails, subsequent processes are blocked.) • try (The system starts to delete the object. If the deletion fails, no deletion retry is performed, and subsequent processes are not blocked.) • false or skip (The object is not deleted. These are the default value options.)

Parameter	Mandatory	Type	Description
delete_eni	No	String	<p>Whether to delete ENI ports (native elastic network interface).</p> <p>Value options:</p> <ul style="list-style-type: none"> • true or block (The system starts to delete the object. If the deletion fails, subsequent processes are blocked.) • try (The system starts to delete the object. If the deletion fails, no deletion retry is performed, and subsequent processes are not blocked.) • false or skip (Skip the deletion.)
delete_efs	No	String	<p>Whether to delete EFS disks.</p> <p>Value options:</p> <ul style="list-style-type: none"> • true or block (The system starts to delete the object. If the deletion fails, subsequent processes are blocked.) • try (The system starts to delete the object. If the deletion fails, no deletion retry is performed, and subsequent processes are not blocked.) • false or skip (The object is not deleted. These are the default value options.)

Parameter	Mandatory	Type	Description
delete_net	No	String	<p>Whether to delete cluster Service and ingress resources, such as ELB load balancers.</p> <p>Value options:</p> <ul style="list-style-type: none"> • true or block (The system starts to delete the object. If the deletion fails, subsequent processes are blocked.) • try (The system starts to delete the object. If the deletion fails, no deletion retry is performed, and subsequent processes are not blocked.) • false or skip (Skip the deletion.)
delete_obs	No	String	<p>Whether to delete OBS volumes.</p> <p>Value options:</p> <ul style="list-style-type: none"> • true or block (The system starts to delete the object. If the deletion fails, subsequent processes are blocked.) • try (The system starts to delete the object. If the deletion fails, no deletion retry is performed, and subsequent processes are not blocked.) • false or skip (The object is not deleted. These are the default value options.)

Parameter	Mandatory	Type	Description
delete_sfs	No	String	<p>Whether to delete SFS volumes.</p> <p>Value options:</p> <ul style="list-style-type: none"> • true or block (The system starts to delete the object. If the deletion fails, subsequent processes are blocked.) • try (The system starts to delete the object. If the deletion fails, no deletion retry is performed, and subsequent processes are not blocked.) • false or skip (The object is not deleted. These are the default value options.)
delete_sfs30	No	String	<p>Whether to delete an SFS 3.0 volume.</p> <p>Example value:</p> <ul style="list-style-type: none"> • true or block (The system starts to delete the object. If the deletion fails, subsequent processes are blocked.) • try (The system starts to delete the object. If the deletion fails, no deletion retry is performed, and subsequent processes are not blocked.) • false or skip (The object is not deleted. These are the default value options.)

Parameter	Mandatory	Type	Description
lts_reclaim_policy	No	String	<p>Whether to delete an LTS log stream.</p> <p>Options:</p> <ul style="list-style-type: none"> • true or block (The system starts to delete the object. If the deletion fails, subsequent processes are blocked.) • try (The system starts to delete the object. If the deletion fails, no deletion retry is performed, and subsequent processes will proceed.) • false or skip (The deletion is skipped. This is the default option.)
tobedeleted	No	String	<p>Whether to use the preset deletion mode for yearly/monthly-billed clusters. This parameter is valid only for yearly/monthly-billed clusters.</p> <p>This parameter must be used together with other deletion parameters. You can specify parameter values or the system uses the default ones.</p> <p>If this parameter is used, the cluster does not delete resources. All query parameters of this request will be preset in the cluster database for identifying the resources to be deleted when a yearly/monthly-billed cluster is unsubscribed from.</p> <p>This request can be executed for multiple times. Each request overwrites the deletion parameters preset last time.</p> <p>Value options:</p> <ul style="list-style-type: none"> • true (preset mode. Only the query parameters are preset and the deletion is not performed.)

Parameter	Mandatory	Type	Description
ondemand_node_policy	No	String	<p>Policies for processing all pay-per-use nodes in a cluster.</p> <p>Options:</p> <ul style="list-style-type: none"> • delete: The nodes will be deleted. • reset: The nodes will be reset and retained. The data stored in the nodes will not be retained. • retain: The nodes and the data stored in the nodes will be retained. The nodes will not be reset.
periodic_node_policy	No	String	<p>Policies for processing all yearly/monthly nodes in a cluster.</p> <p>Options:</p> <ul style="list-style-type: none"> • reset: The nodes will be reset and retained. The data stored in the nodes will not be retained. • retain: The nodes and the data stored in the nodes will be retained. The nodes will not be reset.

Request Parameters

Table 4-113 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	<p>Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token.</p>

Response Parameters

Status code: 200

Table 4-114 Response body parameters

Parameter	Type	Description
kind	String	API type. The value is fixed at Cluster or cluster and cannot be changed.
apiVersion	String	API version. The value is fixed at v3 and cannot be changed.
metadata	ClusterMetadata object	Basic information about a cluster. Metadata is a collection of attributes.
spec	ClusterSpec object	Detailed description of the cluster. CCE creates or updates objects by defining or updating spec .
status	ClusterStatus object	Cluster status and job ID of the cluster creation job.

Table 4-115 ClusterMetadata

Parameter	Type	Description
name	String	Cluster name. Enter 4 to 128 characters, starting with a lowercase letter and not ending with a hyphen (-). Only lowercase letters, digits, and hyphens (-) are allowed.
uid	String	Cluster ID, which uniquely identifies a cluster. This ID is automatically generated after a cluster is created. Only the automatically generated ID will take effect. When you create a yearly/monthly cluster, no cluster ID will be returned in the response body.

Parameter	Type	Description
alias	String	<p>Alias of a cluster name displayed on the CCE console, and the name can be changed.</p> <p>Enter 4 to 128 characters, starting with a lowercase letter and not ending with a hyphen (-). Only lowercase letters, digits, and hyphens (-) are allowed.</p> <p>A cluster alias must be unique.</p> <p>In the request body for creating or updating a cluster, if the cluster alias is not specified or set to null, the cluster name will be used as the cluster alias. In the response body for obtaining a cluster, the cluster alias is returned. If it is not configured, the cluster name will be returned.</p>
annotations	Map<String,String>	<p>Cluster annotations, in the format of key-value pairs.</p> <pre>"annotations": { "key1": "value1", "key2": "value2" }</pre> <p>NOTE</p> <ul style="list-style-type: none"> • annotations: Does not label or select objects. The metadata in annotations may be small or large, structured or unstructured, and may include characters that are not allowed in labels. • This field is not stored in the database and is used only to specify the add-ons to be installed in the cluster. • Install ICAgent during cluster creation by adding the key-value pair "cluster.install.addons.external/install":{"addonTemplateName":"icagent"}".
labels	Map<String,String>	<p>Cluster labels, in the format of key-value pairs.</p> <p>NOTE</p> <p>The value of this field is automatically generated by the system and is used by the frontend to identify the features supported by the cluster during the upgrade. Customized values are invalid.</p>
creationTimestamp	String	Time when the cluster was created.
updateTimestamp	String	Time when the cluster was updated.

Table 4-116 ClusterSpec

Parameter	Type	Description
category	String	<p>Cluster type. Options:</p> <ul style="list-style-type: none"> ● CCE: CCE cluster CCE cluster supports hybrid deployment of VMs and BMSs, and heterogeneous nodes such as GPU and NPU nodes, allowing you to run your containers in a secure and stable container runtime environment based on a high-performance network model. ● Turbo: CCE Turbo cluster One-stop, cost-effective CCE Turbo clusters run on the cloud native 2.0 infrastructure featuring hardware-software synergy for lossless networking, high security and reliability, and intelligent scheduling.
type	String	<p>Master node architecture:</p> <ul style="list-style-type: none"> ● VirtualMachine: x86 ● ARM64: Arm-based Kunpeng

Parameter	Type	Description
flavor	String	<p>Cluster specifications. Specifications of clusters 1.15 or later versions can be changed after they are created. For details, see Modifying Cluster Specifications. Options:</p> <ul style="list-style-type: none"> ● cce.s1.small: a small-scale CCE cluster with one master node and a maximum of 50 worker nodes ● cce.s1.medium: a medium-scale CCE cluster with one master node and a maximum of 200 worker nodes ● cce.s2.small: a small-scale CCE cluster with three master nodes and a maximum of 50 worker nodes ● cce.s2.medium: a medium-scale CCE cluster with three master nodes and a maximum of 200 worker nodes ● cce.s2.large: a large-scale CCE cluster with three master nodes and a maximum of 1,000 worker nodes ● cce.s2.xlarge: an ultra-large-scale CCE cluster with three master nodes and a maximum of 2,000 worker nodes <p>NOTE The fields in the parameters are described as follows:</p> <ul style="list-style-type: none"> ● s1: specifies a cluster with one master node. If the master node is faulty, the cluster will become unavailable, but running workloads in the cluster are not affected. ● s2: specifies an HA cluster with three master nodes. If one of the master nodes is faulty, the cluster is still available. ● dec: specifies a DeC CCE cluster. For example, cce.dec.s1.small specifies a small-scale, DeC CCE cluster with one master node and a maximum of 50 worker nodes. ● small: specifies that a cluster can manage a maximum of 50 worker nodes. ● medium: specifies that a cluster can manage a maximum of 200 worker nodes. ● large: specifies that a cluster can manage a maximum of 1,000 worker nodes. ● xlarge: specifies that a cluster can manage a maximum of 2,000 worker nodes.

Parameter	Type	Description
version	String	<p>Cluster version, which mirrors the baseline version of the Kubernetes community. The latest version is recommended.</p> <p>You can create clusters of two latest versions on the CCE console. To learn which cluster versions are available, log in to the CCE console, create a cluster, and check the Cluster Version parameter.</p> <p>You can call APIs to create clusters of other versions. However, these cluster versions will be gradually terminated. For details about the support policy, see the CCE announcement.</p> <p>NOTE</p> <ul style="list-style-type: none"> • If not specified, a cluster of the latest version will be created. • If a baseline cluster version is specified but the R version is not specified, a cluster of the latest R version will be created by default. It is a good practice not to specify the R version. • CCE Turbo clusters of v1.19 or later are commercially available.
platformVersion	String	<p>CCE cluster platform version, indicating the internal version under the cluster version (version). Platform versions are used to trace iterations in a major cluster version. They are unique within a major cluster version and recounted when the major cluster version changes. This parameter cannot be customized. When you create a cluster, the latest corresponding platform version is automatically selected.</p> <p>The format of platformVersion is cce.X.Y.</p> <ul style="list-style-type: none"> • X: internal feature version, indicating changes in features, patches, or OS support in the cluster version. The value starts from 1 and increases monotonically. • Y: patch version of an internal feature version. It is used only for software package update after the feature version is brought online. No other modification is involved. The value starts from 0 and increases monotonically.

Parameter	Type	Description
description	String	Cluster description, for example, which purpose the cluster is intended to serve. By default, this field is left unspecified. To modify cluster description after the cluster is created, call the API for updating cluster information or go to the cluster details page on the CCE console. Only UTF-8 encoding is supported.
customSan	Array of strings	Custom SAN field in the server certificate of the cluster API server, which must comply with the SSL and X509 format specifications. <ol style="list-style-type: none"> 1. Duplicate names are not allowed. 2. Must comply with the IP address and domain name formats. Example: SAN 1: DNS Name=example.com SAN 2: DNS Name=www.example.com SAN 3: DNS Name=example.net SAN 4: IP Address=93.184.216.34
ipv6enable	Boolean	Whether the cluster supports IPv6 addresses. This field is supported in clusters of v1.15 and later versions.
hostNetwork	HostNetwork object	Node networking parameters, including VPC and subnet ID. This field is mandatory because nodes in a cluster communicate with each other by using a VPC.
containerNetwork	ContainerNetwork object	Container networking parameters, including the container network model and container CIDR block.
eniNetwork	EniNetwork object	Configuration of the Cloud Native Network 2.0 model. Specify this field when creating a CCE Turbo cluster.
serviceNetwork	ServiceNetwork object	Service CIDR block, including IPv4 CIDR blocks.
authentication	Authentication object	Configurations of the cluster authentication mode.
billingMode	Integer	Billing mode of a cluster. <ul style="list-style-type: none"> • 0: pay-per-use • 1: yearly/monthly Defaults to pay-per-use.
masters	Array of MasterSpec objects	Advanced configurations of master nodes

Parameter	Type	Description
kubernetesSvcIpRange	String	Service CIDR blocks for Kubernetes clusterIPs. This field is available only for clusters of v1.11.7 and later. If this parameter is not specified during cluster creation, the default value 10.247.0.0/16 will be used. This parameter is deprecated. Use serviceNetwork instead. The new field contains the IPv4 CIDR blocks.
clusterTags	Array of ResourceTag objects	Cluster resource tags.
kubeProxyMode	String	Service forwarding mode. Options: <ul style="list-style-type: none"> • iptables: Traditional kube-proxy uses iptables rules to implement Service load balancing. In this mode, too many iptables rules will be generated when many Services are deployed. In addition, non-incremental updates will cause latency and even tangible performance issues in the case of service traffic spikes. • ipvs: Optimized kube-proxy mode with higher throughput and faster speed. This mode supports incremental updates and can keep connections uninterrupted during Service updates. It is suitable for large-sized clusters. <p>NOTE iptables is used by default.</p>
az	String	AZ. This field is returned only for a query. For details about AZs supported by CCE, see Regions and Endpoints .
extendParam	ClusterExtendedParam object	Extended field to decide whether the cluster will span across AZs or belong to a specified enterprise project, or whether a dedicated CCE cluster is to be created.
supportIstio	Boolean	Whether Istio is supported.
configurationsOverride	Array of PackageConfiguration objects	Cluster default component configuration override. If you specify a component or parameter that is not supported, the configuration item will be ignored. For details about the supported components and their parameters, see Cluster Configuration Management .

Table 4-117 HostNetwork

Parameter	Type	Description
vpc	String	<p>ID of the VPC used to create a master node. You can obtain it in either of the following ways:</p> <ul style="list-style-type: none"> • Method 1: Log in to the VPC console and view the VPC ID in the VPC details page. • Method 2: Use the VPC API to obtain VPC IDs. <p>For details, see Querying VPCs.</p>
subnet	String	<p>Network ID of the subnet used to create a master node. You can obtain it in either of the following ways:</p> <ul style="list-style-type: none"> • Method 1: Log in to the VPC console and click the target subnet on the Subnets page. You can view the network ID on the displayed page. • Method 2: Use the VPC API to obtain subnets. <p>For details, see Querying Subnets.</p>
SecurityGroup	String	<p>Default worker node security group ID of the cluster. If specified, the cluster will be bound to the target security group. Otherwise, the system will automatically create a default worker node security group for you. The default worker node security group needs to allow access from certain ports to ensure normal communications. For details, see How Do I Harden the Automatically Created Security Group Rules for CCE Cluster Nodes.</p>

Table 4-118 ContainerNetwork

Parameter	Type	Description
mode	String	<p>Container network model. Select one of the following possible values:</p> <ul style="list-style-type: none"> • overlay_l2: an overlay_l2 network (container tunnel network) built for containers by using OpenVSwitch (OVS). • vpc-router: an underlay_l2 network built for containers by using IPvlan and custom VPC routes. • eni: Cloud Native Network 2.0. This model has integrated cloud native elastic network interfaces (ENIs), uses VPC CIDR blocks to allocate container IP addresses, and allows direct traffic distribution to containers through a load balancer for high performance. Use this model when creating a CCE Turbo cluster.
cidr	String	<p>Container CIDR block. Recommended: 10.0.0.0/12-19, 172.16.0.0/16-19, or 192.168.0.0/16-19. If the selected CIDR block conflicts with existing ones, an error will be reported.</p> <p>Not editable after the cluster is created. (deprecated. A specified cidrs will make cidr invalid.)</p>
cidrs	Array of ContainerCIDR objects	<p>List of container CIDR blocks. In clusters of v1.21 or later, the cidrs field is used. When the cluster network type is vpc-router, you can configure a maximum of 20 container CIDR blocks. In cluster versions earlier than v1.21, if the cidrs field is used, the first CIDR element in the array is used as the container CIDR block.</p> <p>The configuration cannot be modified after the cluster is created.</p>

Table 4-119 ContainerCIDR

Parameter	Type	Description
cidr	String	<p>Container CIDR block. Recommended: 10.0.0.0/12-19, 172.16.0.0/16-19, and 192.168.0.0/16-19</p>

Table 4-120 EniNetwork

Parameter	Type	Description
eniSubnetId	String	IPv4 subnet ID of ENI subnet. (IPv6 is not supported and is being discarded.) You can obtain it in either of the following ways: <ul style="list-style-type: none"> Method 1: Log in to the VPC console and click the target subnet on the Subnets page. You can view the IPv4 subnet ID on the displayed page. Method 2: Use the VPC API to obtain subnet IDs. For details, see Querying Subnets .
eniSubnetCIDR	String	ENI subnet CIDR (being discarded)
subnets	Array of NetworkSubnet objects	List of IPv4 subnet IDs

Table 4-121 NetworkSubnet

Parameter	Type	Description
subnetID	String	IPv4 subnet ID of the subnet for creating master nodes. IPv6 is not supported. You can obtain it in either of the following ways: <ul style="list-style-type: none"> Method 1: Log in to the VPC console and click the target subnet on the Subnets page. You can view the IPv4 subnet ID on the displayed page. Method 2: Use the VPC API to obtain subnets. For details, see Querying Subnets .

Table 4-122 ServiceNetwork

Parameter	Type	Description
IPv4CIDR	String	Value range of the Kubernetes clusterIP IPv4 CIDR blocks. If this parameter is not specified during cluster creation, the default value 10.247.0.0/16 will be used.

Table 4-123 Authentication

Parameter	Type	Description
mode	String	Cluster authentication mode. <ul style="list-style-type: none"> Clusters of Kubernetes v1.11 or earlier support x509, rbac, and authenticating_proxy. Defaults to x509. Clusters of Kubernetes v1.13 or later support rbac and authenticating_proxy. Defaults to rbac.
authenticatingProxy	AuthenticatingProxy object	Configuration related to the <code>authenticating_proxy</code> mode. This field is mandatory when the authentication mode is authenticating_proxy .

Table 4-124 AuthenticatingProxy

Parameter	Type	Description
ca	String	X509 CA certificate (Base64-encoded) configured in <code>authenticating_proxy</code> mode. This field is mandatory when the cluster authentication mode is authenticating_proxy . Maximum size: 1 MB
cert	String	Client certificate issued by the X509 CA certificate configured in <code>authenticating_proxy</code> mode, which is used for authentication from kube-apiserver to the extended API server. (The value must be Base64-encoded.) This field is mandatory when the cluster authentication mode is authenticating_proxy .
privateKey	String	Private key of the client certificate issued by the X509 CA certificate configured in <code>authenticating_proxy</code> mode, which is used for authentication from kube-apiserver to the extended API server. The private key used by the Kubernetes cluster does not support password encryption. Use an unencrypted private key. (The value must be Base64-encoded.) This field is mandatory when the cluster authentication mode is authenticating_proxy .

Table 4-125 MasterSpec

Parameter	Type	Description
availabilityZone	String	AZ

Table 4-126 ResourceTag

Parameter	Type	Description
key	String	Key. <ul style="list-style-type: none"> • Cannot be null. Max characters: 128. • Use letters, digits, and spaces in UTF-8 format. • Can contain the following special characters: <code>._:/=+-@</code>. • Cannot start with <code>_sys_</code>.
value	String	Value. <ul style="list-style-type: none"> • Can be null but not the default. Max characters: 255. • Use letters, digits, and spaces in UTF-8 format. • Can contain the following special characters: <code>._:/=+-@</code>.

Table 4-127 ClusterExtendParam

Parameter	Type	Description
clusterAZ	String	AZ of master nodes in a cluster. For details about AZs supported by CCE, see Regions and Endpoints . <ul style="list-style-type: none"> • multi_az: (Optional) The cluster will span across AZs. Multiple AZs can be configured only when a cluster with multiple master nodes is used. • <i>AZ of the dedicated cloud computing pool</i>: The cluster will be deployed in the DeC AZ. It is mandatory for DeC CCE clusters.

Parameter	Type	Description
dssMasterVolumes	String	<p>Whether the system and data disks of a master node use dedicated distributed storage. If this parameter is omitted or left unspecified, EVS disks are used by default.</p> <p>This parameter is mandatory for dedicated CCE clusters. It is in the following format: <code><rootVol.dssPoolID>.<rootVol.volType>;<dataVol.dssPoolID>.<dataVol.volType></code></p> <p>Field description:</p> <ul style="list-style-type: none"> • rootVol is the system disk. dataVol is the data disk. • dssPoolID indicates the ID of the DSS storage pool. • volType indicates the storage volume type of the DSS storage pool, such as SAS and SSD. <p>Example: <code>c950ee97-587c-4f24-8a74-3367e3da570f.sas;6edbc2f4-1507-44f8-ac0d-eed1d2608d38.ssd</code></p> <p>NOTE This field cannot be configured for non-dedicated CCE clusters.</p>
enterpriseProjectId	String	<p>ID of the enterprise project that a cluster belongs to.</p> <p>NOTE</p> <ul style="list-style-type: none"> • An enterprise project can be configured only after the enterprise project function is enabled. • The enterprise project to which the cluster belongs must be the same as that to which other cloud service resources associated with the cluster belong.

Parameter	Type	Description
kubeProxyMode	String	<p>Service forwarding mode. Two modes are available:</p> <ul style="list-style-type: none"> • iptables: Traditional kube-proxy uses iptables rules to implement Service load balancing. In this mode, too many iptables rules will be generated when many Services are deployed. In addition, non-incremental updates will cause latency and even tangible performance issues in the case of service traffic spikes. • ipvs: Optimized kube-proxy mode with higher throughput and faster speed. This mode supports incremental updates and can keep connections uninterrupted during Service updates. It is suitable for large-sized clusters. <p>NOTE This parameter has been deprecated. If this parameter and kubeProxyMode in ClusterSpec are specified at the same time, the latter is used.</p>
clusterExternalIP	String	EIP of the master node
alpha.cce/fixPoolMask	String	<p>Number of mask bits of the fixed IP address pool of the container network model. This field is supported only for the VPC network model (vpc-router).</p> <p>This parameter determines the number of container IP addresses that can be allocated to a node. The maximum number of pods that can be created on a node is decided by this parameter and maxPods set during node creation.</p> <p>For details, see Maximum Number of Pods That Can Be Created on a Node.</p> <p>For integer characters, the value ranges from 24 to 28.</p>
decMasterFlavor	String	Specifications of the master node in the dedicated hybrid cluster.
dockerUmaskMode	String	Default UmaskMode configuration of Docker in a cluster. The value can be secure or normal . If this parameter is not specified, normal is used by default.

Parameter	Type	Description
kubernetes.io/cpuManagerPolicy	String	<p>Cluster CPU management policy. The value can be none (or null) or static. The default value is none (or null).</p> <ul style="list-style-type: none"> ● none or null: disables pods from exclusively occupying CPUs. Select this option if you want a large pool of shareable CPU cores. ● static: enables pods to exclusively occupy CPUs. Select this option if your workload is sensitive to CPU cache and scheduling latency. In a CCE Turbo cluster, this setting is valid only for nodes where common containers, not Kata containers, run.
orderId	String	<p>Order ID. This parameter is returned in the response when the cluster is billed on a yearly/monthly basis with auto payment enabled (only in creation scenarios).</p>
periodType	String	<ul style="list-style-type: none"> ● month: The unit is month. ● year: The unit is year. <p>NOTE Request parameter, which is valid and mandatory when billingMode is set to 1 (yearly/monthly billing). Response parameter, which is returned only when a yearly/monthly cluster is created.</p>
periodNum	Integer	<p>Subscription duration. The value can be:</p> <ul style="list-style-type: none"> ● If periodType is month, the value ranges from 1 to 9. ● If periodType is year, the value ranges from 1 to 3. <p>NOTE Request parameter, which is valid and mandatory when billingMode is set to 1. Response parameter, which is returned only when a yearly/monthly cluster is created.</p>
isAutoRenew	String	<p>Whether auto renewal is enabled.</p> <ul style="list-style-type: none"> ● true: Auto renewal is enabled. ● false: Auto renewal is not enabled. <p>NOTE This field is valid when billingMode is set to 1. If not specified, auto renewal is not enabled.</p>

Parameter	Type	Description
isAutoPay	String	Whether to deduct fees automatically. <ul style="list-style-type: none"> • true: Enable automatic fee deduction. • false: Do not enable automatic fee deduction. <p>NOTE This field is valid when billingMode is set to 1. If not specified, auto fee deduction is not enabled.</p>
upgradefrom	String	Records of how the cluster is upgraded to the current version

Table 4-128 PackageConfiguration

Parameter	Type	Description
name	String	Component name.
configurations	Array of ConfigurationItem objects	Component configuration item.

Table 4-129 ConfigurationItem

Parameter	Type	Description
name	String	Component configuration item name.
value	Object	Component configuration item value.

Table 4-130 ClusterStatus

Parameter	Type	Description
phase	String	<p>Cluster status. Options:</p> <ul style="list-style-type: none"> ● Available: The cluster is running properly. ● Unavailable: The cluster is exhibiting unexpected behavior. Manually delete it. ● ScalingUp: Nodes are being added to the cluster. ● ScalingDown: The cluster is being downsized to fewer nodes. ● Creating: The cluster is being created. ● Deleting: The cluster is being deleted. ● Upgrading: The cluster is being upgraded. ● Resizing: Cluster specifications are being changed. ● RollingBack: The cluster is being rolled back. ● RollbackFailed: The cluster rollback is abnormal. ● Hibernating: The cluster is being hibernated. ● Hibernation: The cluster is in hibernation. ● Awaking: The cluster is being woken up from hibernation. ● Empty: The cluster does not have any resources. This field is discarded. ● Error: Resources in the cluster are abnormal. Manually delete the cluster.
jobID	String	<p>ID of the task associated with the cluster in the current state. Options:</p> <ul style="list-style-type: none"> ● ID of the associated task returned when creating a cluster. You can use it to obtain the auxiliary tasks for creating a cluster. ● ID of the associated task returned when a cluster fails to be deleted or is deleted. If this parameter is not empty, you can use the task ID to obtain the auxiliary tasks for deleting a cluster. <p>NOTE Tasks are short-lived. Do not use task information in scenarios such as cluster status determination.</p>
reason	String	Reason of cluster state change. This parameter is returned if the cluster is not in the Available state.

Parameter	Type	Description
message	String	Detailed information about why the cluster changes to the current state. This parameter is returned if the cluster is not in the Available state.
endpoints	Array of ClusterEndpoints objects	Access address of kube-apiserver in the cluster.
isLocked	Boolean	The CBC resource is locked.
lockScene	String	Scenario where the CBC resource is locked.
lockSource	String	Resource locking.
lockSourceId	String	ID of the locked resource.
deleteOption	Object	Whether to delete configurations. This parameter is contained only in the response to the deletion request.
deleteStatus	Object	Whether to delete the status information. This parameter is contained only in the response to the deletion request.

Table 4-131 ClusterEndpoints

Parameter	Type	Description
url	String	Access address of kube-apiserver in the cluster.
type	String	Type of the cluster access address. <ul style="list-style-type: none"> • Internal: address for internal network access • External: address for external network access

Example Requests

None

Example Responses

Status code: 200

The job for deleting a cluster is successfully delivered.

```
{
  "kind": "Cluster",
  "apiVersion": "v3",
  "metadata": {
```

```
"alias" : "mycluster",
"name" : "mycluster",
"uid" : "fc563b3c-9552-11e8-8beb-0255ac106311",
"creationTimestamp" : "2018-08-01 06:20:28.81667161 +0000 UTC",
"updateTimestamp" : "2018-08-01 09:23:38.944333282 +0000 UTC"
},
"spec" : {
  "type" : "VirtualMachine",
  "flavor" : "cce.s1.small",
  "version" : "v1.7.3-r13",
  "description" : "new description",
  "hostNetwork" : {
    "vpc" : "cbed56e8-03e7-4304-a477-b54bef0857c3",
    "subnet" : "5de50062-2be2-4a52-893e-e0906e3e9c9d"
  },
  "containerNetwork" : {
    "mode" : "overlay_l2",
    "cidr" : "172.16.0.0/16"
  },
  "authentication" : {
    "mode" : "x509",
    "authenticatingProxy" : { }
  },
  "billingMode" : 0
},
"status" : {
  "phase" : "Available",
  "jobID" : "e8ebf96c-956d-11e8-a949-0255ac10575d",
  "endpoints" : [ {
    "url" : "https://192.168.0.16:5443",
    "type" : "Internal"
  } ]
}
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class DeleteClusterSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
```

```
.withSk(sk);

CceClient client = CceClient.newBuilder()
    .withCredential(auth)
    .withRegion(CceRegion.valueOf("<YOUR REGION>"))
    .build();
DeleteClusterRequest request = new DeleteClusterRequest();
request.withClusterId("{cluster_id}");
try {
    DeleteClusterResponse response = client.deleteCluster(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteClusterRequest()
        request.cluster_id = "{cluster_id}"
        response = client.delete_cluster(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
```



```

cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteClusterRequest{}
    request.ClusterId = "{cluster_id}"
    response, err := client.DeleteCluster(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The job for deleting a cluster is successfully delivered.

Error Codes

See [Error Codes](#).

4.2.6 Hibernating a Cluster

Function

This API is used to hibernate a running cluster. After a cluster is hibernated, master node resources stop being billed.

Constraints

1. After a cluster is hibernated, resources such as workloads cannot be created or managed in the cluster.
2. After a pay-per-use cluster is hibernated, master node resources are not billed. Other resources such as the nodes to which the cluster belongs, bound EIP, and bandwidth are billed based on their billing mode (yearly/monthly or pay-per-use).

Calling Method

For details, see [Calling APIs](#).

URI

POST /api/v3/projects/{project_id}/clusters/{cluster_id}/operation/hibernate

Table 4-132 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-133 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

None

Example Requests

None

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class HibernateClusterSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        HibernateClusterRequest request = new HibernateClusterRequest();
        request.withClusterId("{cluster_id}");
        try {
            HibernateClusterResponse response = client.hibernateCluster(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
        }
    }
}
```

```
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = HibernateClusterRequest()
        request.cluster_id = "{cluster_id}"
        response = client.hibernate_cluster(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()
```

```
client := cce.NewCceClient(  
    cce.CceClientBuilder().  
        WithRegion(region.ValueOf("<YOUR REGION>")).  
        WithCredential(auth).  
        Build())  
  
request := &model.HibernateClusterRequest{}  
request.ClusterId = "{cluster_id}"  
response, err := client.HibernateCluster(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The cluster hibernation job is successfully delivered. Keep querying the cluster status. When the cluster status changes to Hibernation , the cluster is hibernated.

Error Codes

See [Error Codes](#).

4.2.7 Waking Up a Cluster

Function

This API is used to wake up a hibernated cluster. After the cluster is woken up, master node resources continue to be billed.

Calling Method

For details, see [Calling APIs](#).

URI

POST /api/v3/projects/{project_id}/clusters/{cluster_id}/operation/awake

Table 4-134 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-135 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

None

Example Requests

None

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;
```

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class AwakeClusterSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        AwakeClusterRequest request = new AwakeClusterRequest();
        request.withClusterId("{cluster_id}");
        try {
            AwakeClusterResponse response = client.awakeCluster(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"
```

```
credentials = BasicCredentials(ak, sk, projectId) \

client = CceClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(CceRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = AwakeClusterRequest()
    request.cluster_id = "{cluster_id}"
    response = client.awake_cluster(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AwakeClusterRequest{}
    request.ClusterId = "{cluster_id}"
    response, err := client.AwakeCluster(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The cluster wakeup job is successfully delivered. Keep querying the cluster status. When the cluster status changes to Available , the cluster is woken up successfully.

Error Codes

See [Error Codes](#).

4.2.8 Obtaining a Cluster Certificate

Function

This API is used to obtain a certificate of a specified cluster.

Constraints

This API is applicable to clusters of v1.13 and later.

Calling Method

For details, see [Calling APIs](#).

URI

POST /api/v3/projects/{project_id}/clusters/{cluster_id}/clustercert

Table 4-136 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-137 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Table 4-138 Request body parameters

Parameter	Mandatory	Type	Description
duration	Yes	Integer	Validity period of the cluster certificate. The minimum value is 1 day and the maximum value is 5 years. Therefore, the value ranges from 1 to 1827. (The unit is day. The actual limit depends on the number of leap years in the five years. For example, if there is a leap year in the five years, the upper limit is 1826 days.) If this parameter is set to -1, the maximum value is 5 years.

Response Parameters

Status code: 200

Table 4-139 Response header parameters

Parameter	Type	Description
Port-ID	String	Port ID of the cluster master node

Table 4-140 Response body parameters

Parameter	Type	Description
kind	String	API type. The value is fixed at Config and cannot be changed.
apiVersion	String	API version. The value is fixed at v1 .
preferences	Object	This field is not used currently and is left unspecified by default.
clusters	Array of Clusters objects	Cluster list
users	Array of Users objects	Certificate information and client key information of a specified user
contexts	Array of Contexts objects	Context list
current-context	String	Current context. If publicIp (VM EIP) exists, the value is external . If publicIp does not exist, the value is internal .

Table 4-141 Clusters

Parameter	Type	Description
name	String	Cluster name. <ul style="list-style-type: none"> If publicIp does not exist (that is, no VM EIP exists), there is only one cluster in the cluster list, and the value of this parameter is internalCluster. If publicIp exists (that is, the EIP exists), there are at least two clusters in the cluster list, and the value of this parameter is externalCluster.
cluster	ClusterCert object	Cluster information

Table 4-142 ClusterCert

Parameter	Type	Description
server	String	Server IP address

Parameter	Type	Description
certificate-authority-data	String	Certificate authorization data
insecure-skip-tls-verify	Boolean	Whether to skip the server certificate verification. If the cluster type is externalCluster , the value is true .

Table 4-143 Users

Parameter	Type	Description
name	String	The value is fixed at user .
user	User object	Certificate information and client key information of a specified user

Table 4-144 User

Parameter	Type	Description
client-certificate-data	String	Client certificate
client-key-data	String	PEM encoding data from the TLS client key file

Table 4-145 Contexts

Parameter	Type	Description
name	String	Context name. <ul style="list-style-type: none"> If publicip does not exist (that is, no VM EIP exists), there is only one cluster in the cluster list, and the value of this parameter is internal. If publicip exists (that is, the EIP exists), there are at least two clusters in the cluster list, and the value of this field for all extension contexts is external.
context	Context object	Context information

Table 4-146 Context

Parameter	Type	Description
cluster	String	Cluster context
user	String	User context

Example Requests

Applying for a cluster access certificate valid for 30 days

```
{
  "duration" : 30
}
```

Example Responses

Status code: 200

The certificate of the specified cluster is successfully obtained. For details about the certificate file format, see the Kubernetes v1.Config structure.

```
{
  "kind" : "Config",
  "apiVersion" : "v1",
  "preferences" : { },
  "clusters" : [ {
    "name" : "internalCluster",
    "cluster" : {
      "server" : "https://192.168.1.7:5443",
      "certificate-authority-data" : "Q2VydGhmaWNhdGU6*****FTkQgQ0VSVEIGSUNBVEUtLS0tLQo="
    }
  } ],
  "users" : [ {
    "name" : "user",
    "user" : {
      "client-certificate-data" : "LS0tLS1CRUdJTjBDR*****QVRFLS0tLS0K",
      "client-key-data" : "LS0tLS1CRUdJTj*****BLRVktLS0tLQo="
    }
  } ],
  "contexts" : [ {
    "name" : "internal",
    "context" : {
      "cluster" : "internalCluster",
      "user" : "user"
    }
  } ],
  "current-context" : "internal"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Applying for a cluster access certificate valid for 30 days

```
package com.huaweicloud.sdk.test;
```

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class CreateKubernetesClusterCertSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();

        CreateKubernetesClusterCertRequest request = new CreateKubernetesClusterCertRequest();
        request.withClusterId("{cluster_id}");
        CertDuration body = new CertDuration();
        body.withDuration(30);
        request.withBody(body);
        try {
            CreateKubernetesClusterCertResponse response = client.createKubernetesClusterCert(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Applying for a cluster access certificate valid for 30 days

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
```

example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment

```
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId) \

client = CceClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(CceRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = CreateKubernetesClusterCertRequest()
    request.cluster_id = "{cluster_id}"
    request.body = CertDuration(
        duration=30
    )
    response = client.create_kubernetes_cluster_cert(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

Applying for a cluster access certificate valid for 30 days

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateKubernetesClusterCertRequest{}
    request.ClusterId = "{cluster_id}"
    request.Body = &model.CertDuration{
        Duration: int32(30),
    }
    response, err := client.CreateKubernetesClusterCert(request)
    if err == nil {
```

```

    fmt.Printf("%+v\n", response)
  } else {
    fmt.Println(err)
  }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The certificate of the specified cluster is successfully obtained. For details about the certificate file format, see the Kubernetes v1.Config structure.

Error Codes

See [Error Codes](#).

4.2.9 Modifying Cluster Specifications

Function

This API is used to modify the specifications of a cluster.

NOTE

- For details about constraints, see [Changing Cluster Scale](#).

Calling Method

For details, see [Calling APIs](#).

URI

POST /api/v3/projects/{project_id}/clusters/{cluster_id}/operation/resize

Table 4-147 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Parameter	Mandatory	Type	Description
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-148 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Table 4-149 Request body parameters

Parameter	Mandatory	Type	Description
flavorResize	Yes	String	<p>The target flavor to be changed. You can only adjust the maximum number of worker nodes in a cluster. The number of master nodes is fixed and cannot be altered, and the cluster flavors cannot be downgraded. Suppose the original cluster flavor is cce.s2.medium. In that case, you can only upgrade it to cce.s2.large or higher, but you cannot downgrade it to cce.s2.small or cce.s1.medium.</p> <ul style="list-style-type: none"> • cce.s1.small: a small-scale CCE cluster with one master node and a maximum of 50 worker nodes • cce.s1.medium: a medium-scale CCE cluster with one master node and a maximum of 200 worker nodes • cce.s2.small: a small-scale CCE cluster with three master nodes and a maximum of 50 worker nodes • cce.s2.medium: a medium-scale CCE cluster with three master nodes and a maximum of 200 worker nodes • cce.s2.large: a large-scale CCE cluster with three master nodes and a maximum of 1,000 worker nodes • cce.s2.xlarge: an ultra-large-scale CCE cluster with three master nodes and a maximum of 2,000 worker nodes

Parameter	Mandatory	Type	Description
			<p>NOTE</p> <p>The fields in the parameters are described as follows:</p> <ul style="list-style-type: none"> • s1: specifies a cluster with one master node. If the master node is faulty, the cluster will become unavailable, but running workloads in the cluster are not affected. • s2: specifies an HA cluster with three master nodes. If one of the master nodes is faulty, the cluster is still available. • dec: specifies a DeC CCE cluster. For example, cce.dec.s1.small specifies a small-scale, DeC CCE cluster with one master node and a maximum of 50 worker nodes. • small: specifies that a cluster can manage a maximum of 50 worker nodes. • medium: specifies that a cluster can manage a maximum of 200 worker nodes. • large: specifies that a cluster can manage a maximum of 1,000 worker nodes. • xlarge: specifies that a cluster can manage a maximum of 2,000 worker nodes.
extendParam	No	extendParam object	

Table 4-150 extendParam

Parameter	Mandatory	Type	Description
decMasterFlavor	No	String	Specifications of the master node in the dedicated hybrid cluster

Parameter	Mandatory	Type	Description
isAutoPay	No	String	Whether auto payment is enabled. <ul style="list-style-type: none"> • true: Auto payment is enabled. • false: Auto payment is not enabled. NOTE This field is valid for yearly/monthly clusters. If not specified, auto payment is not enabled.

Response Parameters

Status code: 201

Table 4-151 Response body parameters

Parameter	Type	Description
jobID	String	ID of the job
orderID	String	ID of the order for modifying the specifications of a yearly/monthly billed cluster

Example Requests

- Modifying the specifications of the yearly/monthly cluster (automatic payment)

```
POST /api/v3/projects/{project_id}/clusters/{cluster_id}/operation/resize
```

```
{
  "flavorResize": "cce.s1.medium",
  "extendParam": {
    "isAutoPay": "true"
  }
}
```

- Modifying the specifications of the pay-per-use cluster

```
POST /api/v3/projects/{project_id}/clusters/{cluster_id}/operation/resize
```

```
{
  "flavorResize": "cce.s1.medium"
}
```

Example Responses

Status code: 201

The job for modifying the specifications of the pay-per-use cluster is delivered.

```
{  
  "jobID" : "13b8d958-8fcf-11ed-ae3-0255ac1001bd"  
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

- Modifying the specifications of the yearly/monthly cluster (automatic payment)

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cce.v3.region.CceRegion;  
import com.huaweicloud.sdk.cce.v3.*;  
import com.huaweicloud.sdk.cce.v3.model.*;  
  
public class ResizeClusterSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before  
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local  
        // environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        CceClient client = CceClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ResizeClusterRequest request = new ResizeClusterRequest();  
        request.withClusterId("{cluster_id}");  
        ResizeClusterRequestBody body = new ResizeClusterRequestBody();  
        ResizeClusterRequestBodyExtendParam extendParambody = new  
        ResizeClusterRequestBodyExtendParam();  
        extendParambody.withIsAutoPay("true");  
        body.withExtendParam(extendParambody);  
        body.withFlavorResize("cce.s1.medium");  
        request.withBody(body);  
        try {  
            ResizeClusterResponse response = client.resizeCluster(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
        }  
    }  
}
```

```
        System.out.println(e.getErrorMsg());
    }
}
}
```

- Modifying the specifications of the pay-per-use cluster

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ResizeClusterSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        ResizeClusterRequest request = new ResizeClusterRequest();
        request.withClusterId("{cluster_id}");
        ResizeClusterRequestBody body = new ResizeClusterRequestBody();
        body.withFlavorResize("cce.s1.medium");
        request.withBody(body);
        try {
            ResizeClusterResponse response = client.resizeCluster(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- Modifying the specifications of the yearly/monthly cluster (automatic payment)

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ResizeClusterRequest()
        request.cluster_id = "{cluster_id}"
        extendParambody = ResizeClusterRequestBodyExtendParam(
            is_auto_pay="true"
        )
        request.body = ResizeClusterRequestBody(
            extend_param=extendParambody,
            flavor_resize="cce.s1.medium"
        )
        response = client.resize_cluster(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- **Modifying the specifications of the pay-per-use cluster**

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ResizeClusterRequest()
        request.cluster_id = "{cluster_id}"
```

```

request.body = ResizeClusterRequestBody(
    flavor_resize="cce.s1.medium"
)
response = client.resize_cluster(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)

```

Go

- Modifying the specifications of the yearly/monthly cluster (automatic payment)

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ResizeClusterRequest{}
    request.ClusterId = "{cluster_id}"
    isAutoPayExtendParam := "true"
    extendParambody := &model.ResizeClusterRequestBodyExtendParam{
        IsAutoPay: &isAutoPayExtendParam,
    }
    request.Body = &model.ResizeClusterRequestBody{
        ExtendParam: extendParambody,
        FlavorResize: "cce.s1.medium",
    }
    response, err := client.ResizeCluster(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

- Modifying the specifications of the pay-per-use cluster


```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ResizeClusterRequest{}
    request.ClusterId = "{cluster_id}"
    request.Body = &model.ResizeClusterRequestBody{
        FlavorResize: "cce.s1.medium",
    }
    response, err := client.ResizeCluster(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
201	The job for modifying the specifications of the pay-per-use cluster is delivered.

Error Codes

See [Error Codes](#).

4.2.10 Querying a Job

Function

This API is used to obtain information about a job via the job ID returned after a query request is delivered.

NOTE

- The URL for cluster management is in the format of **https://Endpoint/uri**. In the URL, **uri** indicates the resource path, that is, the path for API access.
- You can call this API when:
 - Creating or deleting a cluster
 - Creating or deleting a node

Calling Method

For details, see [Calling APIs](#).

URI

GET /api/v3/projects/{project_id}/jobs/{job_id}

Table 4-152 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
job_id	Yes	String	Job ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-153 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-154 Response body parameters

Parameter	Type	Description
kind	String	API type. The value is fixed at Job and cannot be changed.
apiVersion	String	API version. The value is fixed at v3 and cannot be changed.
metadata	JobMetadata object	Job metadata
spec	JobSpec object	Detailed job parameters
status	JobStatus object	Job status

Table 4-155 JobSpec

Parameter	Type	Description
type	String	Job type. An example value is CreateCluster .
clusterUID	String	ID of the cluster where the job runs.
resourceID	String	ID of the resource on which the job is executed.
resourceName	String	Name of the resource on which the job is executed.
extendParam	Map<String,String>	Extended parameters

Parameter	Type	Description
subJobs	Array of Job objects	Subjob list. <ul style="list-style-type: none"> The list contains details about all subjobs. Generally, a cluster/node creation job consists of multiple subjobs. The job is complete only after all subjobs are complete.

Table 4-156 Job

Parameter	Type	Description
kind	String	API type. The value is fixed at Job and cannot be changed.
apiVersion	String	API version. The value is fixed at v3 and cannot be changed.
metadata	JobMetadata object	Job metadata
spec	JobSpec object	Detailed job parameters
status	JobStatus object	Job status

Table 4-157 JobMetadata

Parameter	Type	Description
uid	String	Job ID
creationTimes tamp	String	Time when the job was created
updateTimest amp	String	Time when the job was updated

Table 4-158 JobStatus

Parameter	Type	Description
phase	String	Job status. Possible values: <ul style="list-style-type: none"> ● JobPhaseInitializing JobPhase = "Initializing" ● JobPhaseRunning JobPhase = "Running" ● JobPhaseFailed JobPhase = "Failed" ● JobPhaseSuccess JobPhase = "Success"
reason	String	Reason why the job is in the current state.

Example Requests

None

Example Responses

Status code: 200

The progress of the specified job is successfully obtained.

```
{
  "kind": "Job",
  "apiVersion": "v3",
  "metadata": {
    "uid": "354331b2c-229a-11e8-9c75-0255ac100ceb",
    "creationTimestamp": "2018-08-02 08:12:40.672772389 +0000 UTC",
    "updateTimestamp": "2018-08-02 08:21:50.478108569 +0000 UTC"
  },
  "spec": {
    "type": "CreateCluster",
    "clusterUID": "4d1ecb2c-229a-11e8-9c75-0255ac100ceb",
    "resourceID": "6f4dcb2c-229a-11e8-9c75-0255ac100ceb",
    "resourceName": "cluster-name",
    "extendParam": {
      "serverID": "bc467e3a-2338-11e8-825b-0255ac100c13"
    }
  },
  "subJobs": [ {
    "kind": "Job",
    "apiVersion": "v3",
    "metadata": {
      "uid": "fd474fab-9606-11e8-baa9-0255ac10215d",
      "creationTimestamp": "2018-08-02 03:52:34.615819618 +0000 UTC",
      "updateTimestamp": "2018-08-02 04:05:29.196243031 +0000 UTC"
    },
    "spec": {
      "type": "InstallMaster",
      "clusterUID": "fcc72de0-9606-11e8-baa8-0255ac10215d",
      "resourceID": "fd3b4ac0-9606-11e8-baa8-0255ac10215d",
      "extendParam": {
        "serverID": "fd3b4ac0-9606-11e8-baa8-0255ac10215d"
      }
    }
  },
  "status": {
    "phase": "Success"
  }
}, {
  "kind": "Job",
  "apiVersion": "v3",
  "metadata": {
```

```
"uid" : "fd474f82-9606-11e8-baa8-0255ac10215d",
"creationTimestamp" : "2018-08-02 03:52:33.859150791 +0000 UTC",
"updateTimestamp" : "2018-08-02 03:52:34.615655429 +0000 UTC"
},
"spec" : {
  "type" : "CreatePSMCert",
  "clusterUID" : "fcc72de0-9606-11e8-baa8-0255ac10215d"
},
"status" : {
  "phase" : "Success"
}
}
]
},
"status" : {
  "phase" : "Running",
  "reason" : ""
}
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ShowJobSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowJobRequest request = new ShowJobRequest();
        request.withJobId("{job_id}");
        try {
            ShowJobResponse response = client.showJob(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
        }
    }
}
```

```
e.printStackTrace();
System.out.println(e.getStatusCode());
System.out.println(e.getRequestId());
System.out.println(e.getErrorCode());
System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowJobRequest()
        request.job_id = "{job_id}"
        response = client.show_job(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
```

```

WithAk(ak).
WithSk(sk).
WithProjectId(projectId).
Build()

client := cce.NewCceClient(
    cce.CceClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowJobRequest{}
request.JobId = "{job_id}"
response, err := client.ShowJob(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The progress of the specified job is successfully obtained.

Error Codes

See [Error Codes](#).

4.2.11 Binding/Unbinding Public API Server Address

Function

This API is used to bind or unbind the public API server address of the cluster by cluster ID.

NOTE

The URL for cluster management is in the format of https://Endpoint/uri. In the URL, **uri** indicates the resource path, that is, the path for API access.

Calling Method

For details, see [Calling APIs](#).

URI

PUT /api/v3/projects/{project_id}/clusters/{cluster_id}/mastereip

Table 4-159 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-160 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Table 4-161 Request body parameters

Parameter	Mandatory	Type	Description
spec	Yes	MasterEIPRequestSpec object	Parameters in the request for binding or unbinding the public APIServer address of a cluster

Table 4-162 MasterEIPRequestSpec

Parameter	Mandatory	Type	Description
action	No	String	Binding or unbinding. Mandatory. <ul style="list-style-type: none"> Binding: The value is fixed at {"action":"bind"}. Unbinding: The value is fixed at {"action":"unbind"}.
spec	No	spec object	Configuration attributes of the elastic IP address to be bound
bandwidth	No	String	Bandwidth (This field has expired and is not recommended.)
elasticip	No	String	ENI IP (This field has expired and is not recommended.)

Table 4-163 spec

Parameter	Mandatory	Type	Description
id	No	String	Specifies the ENI ID. This parameter is mandatory for binding an ENI and is invalid for unbinding an ENI.

Response Parameters

Status code: 200

Table 4-164 Response body parameters

Parameter	Type	Description
metadata	Metadata object	Basic information about the object. Metadata is a collection of attributes.
spec	MasterEIPResponseSpec object	Configuration of the bound public APIServer address of a cluster
status	status object	Status information

Table 4-165 Metadata

Parameter	Type	Description
uid	String	Unique ID.
name	String	Resource name
labels	Map<String,String>	Resource labels in key-value pairs. This is a reserved field and does not take effect.
annotations	Map<String,String>	Resource annotations in the format of key-value pairs.
updateTimestamp	String	Update time.
creationTimestamp	String	Creation time.

Table 4-166 MasterEIPResponseSpec

Parameter	Type	Description
action	String	A binding operation
spec	spec object	Configuration attributes of the elastic IP address to be bound
elasticIp	String	EIP

Table 4-167 spec

Parameter	Type	Description
id	String	ENI ID
eip	EipSpec object	EIP details
isDynamic	Boolean	Dynamic provisioning or not

Table 4-168 EipSpec

Parameter	Type	Description
bandwidth	bandwidth object	Bandwidth information

Table 4-169 bandwidth

Parameter	Type	Description
size	Integer	Bandwidth size
sharetype	String	Bandwidth type

Table 4-170 status

Parameter	Type	Description
privateEndpoint	String	Private IP for accessing the cluster (VIP in the case of an HA cluster)
publicEndpoint	String	Public IP for accessing the cluster

Example Requests

Bind the public API server address to a cluster.

```
{
  "spec" : {
    "action" : "bind",
    "spec" : {
      "id" : "a757a69e-f920-455a-b1ba-d7a22db0fd50"
    }
  }
}
```

Example Responses

Status code: 200

This indicates that the public API server address of the cluster is successfully bound. No response body is returned if the unbinding is successful.

```
{
  "metadata" : { },
  "spec" : {
    "action" : "bind",
    "spec" : {
      "id" : "a757a69e-f920-455a-b1ba-d7a22db0fd50",
      "eip" : {
        "bandwidth" : {
          "size" : 5,
          "sharetype" : "PER"
        }
      }
    },
    "isDynamic" : false
  },
  "elasticIp" : "8.8.8.8"
},
"status" : {
  "privateEndpoint" : "https://192.168.3.238:5443",
  "publicEndpoint" : "https://8.8.8.8:5443"
}
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Bind the public API server address to a cluster.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class UpdateClusterEipSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();

        UpdateClusterEipRequest request = new UpdateClusterEipRequest();
        request.withClusterId("{cluster_id}");
        MasterEIPRequest body = new MasterEIPRequest();
        MasterEIPRequestSpecSpec specSpec = new MasterEIPRequestSpecSpec();
        specSpec.withId("a757a69e-f920-455a-b1ba-d7a22db0fd50");
        MasterEIPRequestSpec specbody = new MasterEIPRequestSpec();
        specbody.withAction(MasterEIPRequestSpec.ActionEnum.fromValue("bind"))
            .withSpec(specSpec);
        body.withSpec(specbody);
        request.withBody(body);
        try {
            UpdateClusterEipResponse response = client.updateClusterEip(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Bind the public API server address to a cluster.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateClusterEipRequest()
        request.cluster_id = "{cluster_id}"
        specSpec = MasterEIPRequestSpecSpec(
            id="a757a69e-f920-455a-b1ba-d7a22db0fd50"
        )
        specbody = MasterEIPRequestSpec(
            action="bind",
            spec=specSpec
        )
        request.body = MasterEIPRequest(
            spec=specbody
        )
        response = client.update_cluster_eip(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Bind the public API server address to a cluster.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
```

```

example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := cce.NewCceClient(
    cce.CceClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateClusterEipRequest{}
request.ClusterId = "{cluster_id}"
idSpec:= "a757a69e-f920-455a-b1ba-d7a22db0fd50"
specSpec := &model.MasterEipRequestSpecSpec{
    Id: &idSpec,
}
actionSpec:= model.GetMasterEipRequestSpecActionEnum().BIND
specbody := &model.MasterEipRequestSpec{
    Action: &actionSpec,
    Spec: specSpec,
}
request.Body = &model.MasterEipRequest{
    Spec: specbody,
}
response, err := client.UpdateClusterEip(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	This indicates that the public API server address of the cluster is successfully bound. No response body is returned if the unbinding is successful.

Error Codes

See [Error Codes](#).

4.2.12 Obtaining Cluster Access Address

Function

This API is used to obtain the cluster access addresses by cluster ID, including the private IP (VIP returned by an HA cluster) and public IP.

NOTE

The URL for cluster management is in the format of https://Endpoint/uri. In the URL, **uri** indicates the resource path, that is, the path for API access.

Calling Method

For details, see [Calling APIs](#).

URI

GET /api/v3/projects/{project_id}/clusters/{cluster_id}/openapi

Table 4-171 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-172 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-173 Response body parameters

Parameter	Type	Description
metadata	Metadata object	Basic information about the object. Metadata is a collection of attributes.
spec	OpenAPISpec object	Parameters for configuring the address for accessing the cluster
status	status object	Status information

Table 4-174 Metadata

Parameter	Type	Description
uid	String	Unique ID.
name	String	Resource name
labels	Map<String,String>	Resource labels in key-value pairs. This is a reserved field and does not take effect.
annotations	Map<String,String>	Resource annotations in the format of key-value pairs.
updateTimestamp	String	Update time.
creationTimestamp	String	Creation time.

Table 4-175 OpenAPISpec

Parameter	Type	Description
spec	spec object	Address for accessing the cluster

Table 4-176 spec

Parameter	Type	Description
eip	EipSpec object	EIP details
isDynamic	Boolean	Dynamic provisioning or not

Table 4-177 EipSpec

Parameter	Type	Description
bandwidth	bandwidth object	Bandwidth information

Table 4-178 bandwidth

Parameter	Type	Description
size	Integer	Bandwidth size
sharetype	String	Bandwidth type

Table 4-179 status

Parameter	Type	Description
privateEndpoint	String	Private IP for accessing the cluster (VIP in the case of an HA cluster)
publicEndpoint	String	Public IP for accessing the cluster

Example Requests

None

Example Responses

Status code: 200

The cluster access address is obtained successfully.

```
{
  "metadata": { },
  "spec": {
    "spec": {
      "eip": {
        "bandwidth": { }
      },
      "IsDynamic": false
    }
  },
  "status": {
    "privateEndpoint": "https://192.168.3.238:5443",
    "publicEndpoint": ""
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ShowClusterEndpointsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowClusterEndpointsRequest request = new ShowClusterEndpointsRequest();
        request.withClusterId("{cluster_id}");
        try {
            ShowClusterEndpointsResponse response = client.showClusterEndpoints(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId) \

client = CceClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(CceRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ShowClusterEndpointsRequest()
    request.cluster_id = "{cluster_id}"
    response = client.show_cluster_endpoints(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowClusterEndpointsRequest{}
    request.ClusterId = "{cluster_id}"
    response, err := client.ShowClusterEndpoints(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The cluster access address is obtained successfully.

Error Codes

See [Error Codes](#).

4.2.13 Obtaining a Cluster's Logging Configurations

Function

This API is used to obtain LTS configurations reported by the cluster components.

Calling Method

For details, see [Calling APIs](#).

URI

GET /api/v3/projects/{project_id}/cluster/{cluster_id}/log-configs

Table 4-180 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Table 4-181 Query Parameters

Parameter	Mandatory	Type	Description
type	No	String	<p>Component type. Options: control, audit, and system-addon. If this parameter is not specified, all component types will be obtained.</p> <ul style="list-style-type: none"> • control: Logs of the components on the master nodes will be obtained. • audit: Audit logs of the master nodes will be obtained. • system-addon: Logs of system add-ons will be obtained.

Request Parameters

Table 4-182 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	<p>Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token.</p>

Response Parameters

Status code: 200

Table 4-183 Response body parameters

Parameter	Type	Description
tll_in_days	Integer	Time for storage
log_configs	Array of log_configs objects	Log configuration items

Table 4-184 log_configs

Parameter	Type	Description
name	String	Log type
enable	Boolean	Whether to collect logs.

Example Requests

The following example describes how to obtain add-on log configurations.

```
/api/v3/projects/{project_id}/cluster/{cluster_id}/log-configs?system-addon
```

Example Responses

Status code: 200

Cluster log configurations are obtained successfully.

```
{
  "log_configs": [ {
    "name": "volcano",
    "enable": true
  }, {
    "name": "coredns",
    "enable": false
  }, {
    "name": "everest",
    "enable": false
  } ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ShowClusterConfigSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";
```

```
ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

CceClient client = CceClient.newBuilder()
    .withCredential(auth)
    .withRegion(CceRegion.valueOf("<YOUR REGION>"))
    .build();
ShowClusterConfigRequest request = new ShowClusterConfigRequest();
request.withClusterId("{cluster_id}");
try {
    ShowClusterConfigResponse response = client.showClusterConfig(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowClusterConfigRequest()
        request.cluster_id = "{cluster_id}"
        response = client.show_cluster_config(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```


Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowClusterConfigRequest{}
    request.ClusterId = "{cluster_id}"
    response, err := client.ShowClusterConfig(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Cluster log configurations are obtained successfully.

Error Codes

See [Error Codes](#).

4.2.14 Configuring Cluster Logs

Function

This API is used to select the master node components whose logs are reported to LTS.

Calling Method

For details, see [Calling APIs](#).

URI

PUT /api/v3/projects/{project_id}/cluster/{cluster_id}/log-configs

Table 4-185 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-186 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Table 4-187 Request body parameters

Parameter	Mandatory	Type	Description
ttl_in_days	No	Integer	Time for storage
log_configs	No	Array of log_configs objects	Log configuration items

Table 4-188 log_configs

Parameter	Mandatory	Type	Description
name	No	String	Log type
enable	No	Boolean	Whether to collect logs.

Response Parameters

Status code: 200

Table 4-189 Response body parameters

Parameter	Type	Description
ttl_in_days	Integer	Time for storage
log_configs	Array of log_configs objects	Log configuration items

Table 4-190 log_configs

Parameter	Type	Description
name	String	Log type
enable	Boolean	Whether to collect logs.

Example Requests

The following example describes how to configure cluster logs that are reported to LTS.

```
/api/v3/projects/{project_id}/cluster/{cluster_id}/log-configs
```

```
{
  "log_configs": [ {
    "name": "kube-apiserver",
```

```
    "enable" : true
  }, {
    "name" : "kube-controller-manager",
    "enable" : false
  }, {
    "name" : "kube-scheduler",
    "enable" : false
  }, {
    "name" : "volcano",
    "enable" : true
  }, {
    "name" : "coredns",
    "enable" : false
  }, {
    "name" : "everest",
    "enable" : false
  }
} ]
}
```

Example Responses

Status code: 200

Cluster logs are configured successfully.

```
{
  "ttl_in_days" : 7,
  "log_configs" : [ {
    "name" : "kube-controller-manager",
    "enable" : true
  }, {
    "name" : "kube-apiserver",
    "enable" : true
  }, {
    "name" : "kube-scheduler",
    "enable" : true
  }, {
    "name" : "audit",
    "enable" : true
  }
]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

The following example describes how to configure cluster logs that are reported to LTS.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateClusterLogConfigSolution {
```

```
public static void main(String[] args) {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running
    // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    String ak = System.getenv("CLOUD_SDK_AK");
    String sk = System.getenv("CLOUD_SDK_SK");
    String projectId = "{project_id}";

    ICredential auth = new BasicCredentials()
        .withProjectId(projectId)
        .withAk(ak)
        .withSk(sk);

    CceClient client = CceClient.newBuilder()
        .withCredential(auth)
        .withRegion(CceRegion.valueOf("<YOUR REGION>"))
        .build();
    UpdateClusterLogConfigRequest request = new UpdateClusterLogConfigRequest();
    request.withClusterId("{cluster_id}");
    ClusterLogConfig body = new ClusterLogConfig();
    List<ClusterLogConfigLogConfigs> listbodyLogConfigs = new ArrayList<>();
    listbodyLogConfigs.add(
        new ClusterLogConfigLogConfigs()
            .withName("kube-apiserver")
            .withEnable(true)
    );
    listbodyLogConfigs.add(
        new ClusterLogConfigLogConfigs()
            .withName("kube-controller-manager")
            .withEnable(false)
    );
    listbodyLogConfigs.add(
        new ClusterLogConfigLogConfigs()
            .withName("kube-scheduler")
            .withEnable(false)
    );
    listbodyLogConfigs.add(
        new ClusterLogConfigLogConfigs()
            .withName("volcano")
            .withEnable(true)
    );
    listbodyLogConfigs.add(
        new ClusterLogConfigLogConfigs()
            .withName("coredns")
            .withEnable(false)
    );
    listbodyLogConfigs.add(
        new ClusterLogConfigLogConfigs()
            .withName("everest")
            .withEnable(false)
    );
    body.withLogConfigs(listbodyLogConfigs);
    request.withBody(body);
    try {
        UpdateClusterLogConfigResponse response = client.updateClusterLogConfig(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

```
}  
}  
}
```

Python

The following example describes how to configure cluster logs that are reported to LTS.

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkcce.v3.region.cce_region import CceRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkcce.v3 import *
```

```
if __name__ == "__main__":
```

```
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.
```

```
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
    ak = __import__('os').getenv("CLOUD_SDK_AK")  
    sk = __import__('os').getenv("CLOUD_SDK_SK")  
    projectId = "{project_id}"
```

```
    credentials = BasicCredentials(ak, sk, projectId) \
```

```
    client = CceClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(CceRegion.value_of("<YOUR REGION>")) \  
        .build()
```

```
try:
```

```
    request = UpdateClusterLogConfigRequest()
```

```
    request.cluster_id = "{cluster_id}"
```

```
    listLogConfigsbody = [  
        ClusterLogConfigLogConfigs(  
            name="kube-apiserver",  
            enable=True
```

```
        ),  
        ClusterLogConfigLogConfigs(  
            name="kube-controller-manager",  
            enable=False
```

```
        ),  
        ClusterLogConfigLogConfigs(  
            name="kube-scheduler",  
            enable=False
```

```
        ),  
        ClusterLogConfigLogConfigs(  
            name="volcano",  
            enable=True
```

```
        ),  
        ClusterLogConfigLogConfigs(  
            name="coredns",  
            enable=False
```

```
        ),  
        ClusterLogConfigLogConfigs(  
            name="everest",  
            enable=False
```

```
        )  
    ]
```

```
    request.body = ClusterLogConfig(  
        log_configs=listLogConfigsbody
```

```
    )  
    response = client.update_cluster_log_config(request)
```

```
    print(response)
```

```
except exceptions.ClientRequestException as e:
```

```
    print(e.status_code)
```

```
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

Go

The following example describes how to configure cluster logs that are reported to LTS.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateClusterLogConfigRequest{}
    request.ClusterId = "{cluster_id}"
    nameLogConfigs := "kube-apiserver"
    enableLogConfigs := true
    nameLogConfigs1 := "kube-controller-manager"
    enableLogConfigs1 := false
    nameLogConfigs2 := "kube-scheduler"
    enableLogConfigs2 := false
    nameLogConfigs3 := "volcano"
    enableLogConfigs3 := true
    nameLogConfigs4 := "coredns"
    enableLogConfigs4 := false
    nameLogConfigs5 := "everest"
    enableLogConfigs5 := false
    var listLogConfigsbody = []model.ClusterLogConfigLogConfigs{
        {
            Name: &nameLogConfigs,
            Enable: &enableLogConfigs,
        },
        {
            Name: &nameLogConfigs1,
            Enable: &enableLogConfigs1,
        },
        {
            Name: &nameLogConfigs2,
            Enable: &enableLogConfigs2,
        },
    }
}
```

```

        Name: &nameLogConfigs3,
        Enable: &enableLogConfigs3,
    },
    {
        Name: &nameLogConfigs4,
        Enable: &enableLogConfigs4,
    },
    {
        Name: &nameLogConfigs5,
        Enable: &enableLogConfigs5,
    },
}
request.Body = &model.ClusterLogConfig{
    LogConfigs: &listLogConfigsbody,
}
response, err := client.UpdateClusterLogConfig(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Cluster logs are configured successfully.

Error Codes

See [Error Codes](#).

4.2.15 Obtaining the Partition List

Function

This API is used to obtain the partition list.

Calling Method

For details, see [Calling APIs](#).

URI

GET /api/v3/projects/{project_id}/clusters/{cluster_id}/partitions

Table 4-191 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

None

Response Parameters

Status code: 200

Table 4-192 Response body parameters

Parameter	Type	Description
kind	String	Resource type
apiVersion	String	API version
items	Array of Partition objects	Cluster partition information

Table 4-193 Partition

Parameter	Type	Description
kind	String	Resource type
apiVersion	String	API version
metadata	metadata object	Metadata of partitions
spec	spec object	Partition configurations

Table 4-194 metadata

Parameter	Type	Description
name	String	Partition name
creationTimes tamp	String	Creation time

Table 4-195 spec

Parameter	Type	Description
hostNetwork	hostNetwork object	Partition subnet
containerNet work	Array of containerNet work objects	Container subnet in the partition
publicBorderG roup	String	Group
category	String	Category

Table 4-196 hostNetwork

Parameter	Type	Description
subnetID	String	Subnet ID

Table 4-197 containerNetwork

Parameter	Type	Description
subnetID	String	Subnet ID

Example Requests

None

Example Responses

Status code: 200

OK

```
{
  "kind" : "List",
  "apiVersion" : "v3",
```

```
"items" : [ {  
  "kind" : "Partition",  
  "apiVersion" : "v3",  
  "metadata" : {  
    "name" : "partitionName",  
    "creationTimestamp" : "2000-1-1 00:00:35.451967 +0000 UTC"  
  },  
  "spec" : {  
    "hostNetwork" : {  
      "subnetID" : "subnetID"  
    },  
    "containerNetwork" : [ {  
      "subnetID" : "subnetID"  
    } ],  
    "publicBorderGroup" : "publicBorderGroup",  
    "category" : "category"  
  }  
}  
]
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cce.v3.region.CceRegion;  
import com.huaweicloud.sdk.cce.v3.*;  
import com.huaweicloud.sdk.cce.v3.model.*;  
  
public class ListPartitionsSolution {  
  
  public static void main(String[] args) {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
    // environment variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running  
    // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    String ak = System.getenv("CLOUD_SDK_AK");  
    String sk = System.getenv("CLOUD_SDK_SK");  
    String projectId = "{project_id}";  
  
    ICredential auth = new BasicCredentials()  
      .withProjectId(projectId)  
      .withAk(ak)  
      .withSk(sk);  
  
    CceClient client = CceClient.newBuilder()  
      .withCredential(auth)  
      .withRegion(CceRegion.valueOf("<YOUR REGION>"))  
      .build();  
    ListPartitionsRequest request = new ListPartitionsRequest();  
    request.withClusterId("{cluster_id}");  
    try {  
      ListPartitionsResponse response = client.listPartitions(request);  
      System.out.println(response.toString());  
    } catch (ConnectionException e) {  
      e.printStackTrace();  
    } catch (RequestTimeoutException e) {  
      e.printStackTrace();  
    }
```

```
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListPartitionsRequest()
        request.cluster_id = "{cluster_id}"
        response = client.list_partitions(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"
```

```

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := cce.NewCceClient(
    cce.CceClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListPartitionsRequest{}
request.ClusterId = "{cluster_id}"
response, err := client.ListPartitions(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	OK

Error Codes

See [Error Codes](#).

4.2.16 Creating a Partition

Function

This API is used to create a partition.

Calling Method

For details, see [Calling APIs](#).

URI

POST /api/v3/projects/{project_id}/clusters/{cluster_id}/partitions

Table 4-198 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-199 Request body parameters

Parameter	Mandatory	Type	Description
kind	No	String	Resource type
apiVersion	No	String	API version
metadata	No	metadata object	Metadata of partitions
spec	No	spec object	Partition configurations

Table 4-200 metadata

Parameter	Mandatory	Type	Description
name	No	String	Partition name

Table 4-201 spec

Parameter	Mandatory	Type	Description
hostNetwork	No	hostNetwork object	Partition subnet
containerNetwork	No	Array of containerNetwork objects	Container subnet in the partition
publicBorderGroup	No	String	Group
category	No	String	Category

Table 4-202 hostNetwork

Parameter	Mandatory	Type	Description
subnetID	No	String	Subnet ID

Table 4-203 containerNetwork

Parameter	Mandatory	Type	Description
subnetID	No	String	Subnet ID

Response Parameters

Status code: 200

Table 4-204 Response body parameters

Parameter	Type	Description
kind	String	Resource type
apiVersion	String	API version
metadata	metadata object	Metadata of partitions
spec	spec object	Partition configurations

Table 4-205 metadata

Parameter	Type	Description
name	String	Partition name
creationTimes tamp	String	Creation time

Table 4-206 spec

Parameter	Type	Description
hostNetwork	hostNetwork object	Partition subnet
containerNet work	Array of containerNet work objects	Container subnet in the partition

Parameter	Type	Description
publicBorderGroup	String	Group
category	String	Category

Table 4-207 hostNetwork

Parameter	Type	Description
subnetID	String	Subnet ID

Table 4-208 containerNetwork

Parameter	Type	Description
subnetID	String	Subnet ID

Example Requests

POST /api/v3/projects/{project_id}/clusters/{cluster_id}/partitions

```
{
  "kind" : "Partition",
  "apiVersion" : "v3",
  "metadata" : {
    "name" : "partitionName"
  },
  "spec" : {
    "hostNetwork" : {
      "subnetID" : "subnetID"
    },
    "containerNetwork" : [ {
      "subnetID" : "subnetID"
    } ],
    "publicBorderGroup" : "publicBorderGroup",
    "category" : "category"
  }
}
```

Example Responses

Status code: 200

OK

```
{
  "kind" : "Partition",
  "apiVersion" : "v3",
  "metadata" : {
    "name" : "partitionName",
    "creationTimestamp" : "2000-1-1 00:00:35.451967 +0000 UTC"
  },
  "spec" : {
    "hostNetwork" : {
```



```
"subnetID" : "subnetID"
},
"containerNetwork" : [ {
  "subnetID" : "subnetID"
} ],
"publicBorderGroup" : "publicBorderGroup",
"category" : "category"
}
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreatePartitionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        CreatePartitionRequest request = new CreatePartitionRequest();
        request.withClusterId("{cluster_id}");
        PartitionReqBody body = new PartitionReqBody();
        List<PartitionReqBodySpecContainerNetwork> listSpecContainerNetwork = new ArrayList<>();
        listSpecContainerNetwork.add(
            new PartitionReqBodySpecContainerNetwork()
                .withSubnetID("subnetID")
        );
        PartitionReqBodySpecHostNetwork hostNetworkSpec = new PartitionReqBodySpecHostNetwork();
        hostNetworkSpec.withSubnetID("subnetID");
        PartitionReqBodySpec specbody = new PartitionReqBodySpec();
        specbody.withHostNetwork(hostNetworkSpec)
            .withContainerNetwork(listSpecContainerNetwork)
            .withPublicBorderGroup("publicBorderGroup")
            .withCategory("category");
        PartitionReqBodyMetadata metadatabody = new PartitionReqBodyMetadata();
        metadatabody.withName("partitionName");
    }
}
```

```
body.withSpec(specbody);
body.withMetadata(metadatabody);
body.withApiVersion("v3");
body.withKind("Partition");
request.withBody(body);
try {
    CreatePartitionResponse response = client.createPartition(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreatePartitionRequest()
        request.cluster_id = "{cluster_id}"
        listContainerNetworkSpec = [
            PartitionReqBodySpecContainerNetwork(
                subnet_id="subnetID"
            )
        ]
        hostNetworkSpec = PartitionReqBodySpecHostNetwork(
            subnet_id="subnetID"
        )
        specbody = PartitionReqBodySpec(
            host_network=hostNetworkSpec,
            container_network=listContainerNetworkSpec,
            public_border_group="publicBorderGroup",
            category="category"
        )
        metadatabody = PartitionReqBodyMetadata(
            name="partitionName"
        )
        request.body = PartitionReqBody(
```

```
        spec=specbody,
        metadata=metadatabody,
        api_version="v3",
        kind="Partition"
    )
    response = client.create_partition(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreatePartitionRequest{}
    request.ClusterId = "{cluster_id}"
    subnetIDContainerNetwork := "subnetID"
    var listContainerNetworkSpec = []model.PartitionReqBodySpecContainerNetwork{
        {
            SubnetID: &subnetIDContainerNetwork,
        },
    }
    subnetIDHostNetwork := "subnetID"
    hostNetworkSpec := &model.PartitionReqBodySpecHostNetwork{
        SubnetID: &subnetIDHostNetwork,
    }
    publicBorderGroupSpec := "publicBorderGroup"
    categorySpec := "category"
    specbody := &model.PartitionReqBodySpec{
        HostNetwork: hostNetworkSpec,
        ContainerNetwork: &listContainerNetworkSpec,
        PublicBorderGroup: &publicBorderGroupSpec,
        Category: &categorySpec,
    }
    nameMetadata := "partitionName"
    metadatabody := &model.PartitionReqBodyMetadata{
```

```
Name: &nameMetadata,  
}  
apiVersionPartitionReqBody:= "v3"  
kindPartitionReqBody:= "Partition"  
request.Body = &model.PartitionReqBody{  
    Spec: specbody,  
    Metadata: metadatabody,  
    ApiVersion: &apiVersionPartitionReqBody,  
    Kind: &kindPartitionReqBody,  
}  
response, err := client.CreatePartition(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	OK

Error Codes

See [Error Codes](#).

4.2.17 Obtaining Partition Details

Function

This API is used to obtain partition details.

Calling Method

For details, see [Calling APIs](#).

URI

GET /api/v3/projects/{project_id}/clusters/{cluster_id}/partitions/{partition_name}

Table 4-209 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
partition_name	Yes	String	Partition name

Request Parameters

None

Response Parameters

Status code: 200

Table 4-210 Response body parameters

Parameter	Type	Description
kind	String	Resource type
apiVersion	String	API version
metadata	metadata object	Metadata of partitions
spec	spec object	Partition configurations

Table 4-211 metadata

Parameter	Type	Description
name	String	Partition name
creationTimestamp	String	Creation time

Table 4-212 spec

Parameter	Type	Description
hostNetwork	hostNetwork object	Partition subnet
containerNetwork	Array of containerNetwork objects	Container subnet in the partition
publicBorderGroup	String	Group
category	String	Category

Table 4-213 hostNetwork

Parameter	Type	Description
subnetID	String	Subnet ID

Table 4-214 containerNetwork

Parameter	Type	Description
subnetID	String	Subnet ID

Example Requests

None

Example Responses

Status code: 200

OK

```
{
  "kind": "Partition",
  "apiVersion": "v3",
  "metadata": {
    "name": "partitionName",
    "creationTimestamp": "2000-1-1 00:00:35.451967 +0000 UTC"
  },
  "spec": {
    "hostNetwork": {
      "subnetID": "subnetID"
    },
    "containerNetwork": [ {
      "subnetID": "subnetID"
    } ],
    "publicBorderGroup": "publicBorderGroup",
    "category": "category"
  }
}
```

```
}  
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cce.v3.region.CceRegion;  
import com.huaweicloud.sdk.cce.v3.*;  
import com.huaweicloud.sdk.cce.v3.model.*;  
  
public class ShowPartitionSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        CceClient client = CceClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ShowPartitionRequest request = new ShowPartitionRequest();  
        request.withPartitionName("{partition_name}");  
        request.withClusterId("{cluster_id}");  
        try {  
            ShowPartitionResponse response = client.showPartition(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

Python

```
# coding: utf-8  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials
```

```
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowPartitionRequest()
        request.partition_name = "{partition_name}"
        request.cluster_id = "{cluster_id}"
        response = client.show_partition(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowPartitionRequest{}
    request.PartitionName = "{partition_name}"
```



```
request.ClusterId = "{cluster_id}"
response, err := client.ShowPartition(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	OK

Error Codes

See [Error Codes](#).

4.2.18 Updating a Partition

Function

This API is used to update a partition.

Calling Method

For details, see [Calling APIs](#).

URI

PUT /api/v3/projects/{project_id}/clusters/{cluster_id}/partitions/{partition_name}

Table 4-215 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Parameter	Mandatory	Type	Description
partition_name	Yes	String	Partition name

Request Parameters

Table 4-216 Request body parameters

Parameter	Mandatory	Type	Description
kind	No	String	Resource type
apiVersion	No	String	API version
metadata	No	metadata object	Metadata of partitions
spec	No	spec object	Partition configurations

Table 4-217 metadata

Parameter	Mandatory	Type	Description
name	No	String	Partition name

Table 4-218 spec

Parameter	Mandatory	Type	Description
hostNetwork	No	hostNetwork object	Partition subnet
containerNetwork	No	Array of containerNetwork objects	Container subnet in the partition
publicBorderGroup	No	String	Group
category	No	String	Category

Table 4-219 hostNetwork

Parameter	Mandatory	Type	Description
subnetID	No	String	Subnet ID

Table 4-220 containerNetwork

Parameter	Mandatory	Type	Description
subnetID	No	String	Subnet ID

Response Parameters

Status code: 200

Table 4-221 Response body parameters

Parameter	Type	Description
kind	String	Resource type
apiVersion	String	API version
metadata	metadata object	Metadata of partitions
spec	spec object	Partition configurations

Table 4-222 metadata

Parameter	Type	Description
name	String	Partition name
creationTimes tamp	String	Creation time

Table 4-223 spec

Parameter	Type	Description
hostNetwork	hostNetwork object	Partition subnet
containerNet work	Array of containerNet work objects	Container subnet in the partition
publicBorderG roup	String	Group
category	String	Category

Table 4-224 hostNetwork

Parameter	Type	Description
subnetID	String	Subnet ID

Table 4-225 containerNetwork

Parameter	Type	Description
subnetID	String	Subnet ID

Example Requests

```
PUT /api/v3/projects/{project_id}/clusters/{cluster_id}/partitions/{partition_name}
{
  "kind": "Partition",
  "apiVersion": "v3",
  "metadata": {
    "name": "partitionName"
  },
  "spec": {
    "hostNetwork": {
      "subnetID": "subnetID"
    },
    "containerNetwork": [ {
      "subnetID": "subnetID"
    } ],
    "publicBorderGroup": "publicBorderGroup",
    "category": "category"
  }
}
```

Example Responses

Status code: 200

OK

```
{
  "kind": "Partition",
  "apiVersion": "v3",
  "metadata": {
    "name": "partitionName",
    "creationTimestamp": "2000-1-1 00:00:35.451967 +0000 UTC"
  },
  "spec": {
    "hostNetwork": {
      "subnetID": "subnetID"
    },
    "containerNetwork": [ {
      "subnetID": "subnetID"
    } ],
    "publicBorderGroup": "publicBorderGroup",
    "category": "category"
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdatePartitionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdatePartitionRequest request = new UpdatePartitionRequest();
        request.withPartitionName("{partition_name}");
        request.withClusterId("{cluster_id}");
        PartitionReqBody body = new PartitionReqBody();
        List<PartitionReqBodySpecContainerNetwork> listSpecContainerNetwork = new ArrayList<>();
        listSpecContainerNetwork.add(
            new PartitionReqBodySpecContainerNetwork()
                .withSubnetID("subnetID")
        );
        PartitionReqBodySpecHostNetwork hostNetworkSpec = new PartitionReqBodySpecHostNetwork();
        hostNetworkSpec.withSubnetID("subnetID");
        PartitionReqBodySpec specbody = new PartitionReqBodySpec();
        specbody.withHostNetwork(hostNetworkSpec)
            .withContainerNetwork(listSpecContainerNetwork)
            .withPublicBorderGroup("publicBorderGroup")
            .withCategory("category");
        PartitionReqBodyMetadata metadatabody = new PartitionReqBodyMetadata();
        metadatabody.withName("partitionName");
        body.withSpec(specbody);
        body.withMetadata(metadatabody);
        body.withApiVersion("v3");
        body.withKind("Partition");
        request.withBody(body);
        try {
            UpdatePartitionResponse response = client.updatePartition(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        }
    }
}
```

```
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdatePartitionRequest()
        request.partition_name = "{partition_name}"
        request.cluster_id = "{cluster_id}"
        listContainerNetworkSpec = [
            PartitionReqBodySpecContainerNetwork(
                subnet_id="subnetID"
            )
        ]
        hostNetworkSpec = PartitionReqBodySpecHostNetwork(
            subnet_id="subnetID"
        )
        specbody = PartitionReqBodySpec(
            host_network=hostNetworkSpec,
            container_network=listContainerNetworkSpec,
            public_border_group="publicBorderGroup",
            category="category"
        )
        metadatabody = PartitionReqBodyMetadata(
            name="partitionName"
        )
        request.body = PartitionReqBody(
            spec=specbody,
            metadata=metadatabody,
            api_version="v3",
            kind="Partition"
        )
        response = client.update_partition(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
```

```
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdatePartitionRequest{}
    request.PartitionName = "{partition_name}"
    request.ClusterId = "{cluster_id}"
    subnetIDContainerNetwork := "subnetID"
    var listContainerNetworkSpec = []model.PartitionReqBodySpecContainerNetwork{
        {
            SubnetID: &subnetIDContainerNetwork,
        },
    },
    subnetIDHostNetwork := "subnetID"
    hostNetworkSpec := &model.PartitionReqBodySpecHostNetwork{
        SubnetID: &subnetIDHostNetwork,
    }
    publicBorderGroupSpec := "publicBorderGroup"
    categorySpec := "category"
    specbody := &model.PartitionReqBodySpec{
        HostNetwork: hostNetworkSpec,
        ContainerNetwork: &listContainerNetworkSpec,
        PublicBorderGroup: &publicBorderGroupSpec,
        Category: &categorySpec,
    }
    nameMetadata := "partitionName"
    metadatabody := &model.PartitionReqBodyMetadata{
        Name: &nameMetadata,
    }
    apiVersionPartitionReqBody := "v3"
    kindPartitionReqBody := "Partition"
    request.Body = &model.PartitionReqBody{
        Spec: specbody,
        Metadata: metadatabody,
        ApiVersion: &apiVersionPartitionReqBody,
```

```
Kind: &kindPartitionReqBody,  
}  
response, err := client.UpdatePartition(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	OK

Error Codes

See [Error Codes](#).

4.3 Node Management

4.3.1 Creating a Node

Function

This API is used to create a node in a specified cluster.

NOTE

- If there is no cluster, [create one](#).
- The URL for cluster management is in the format of **https://Endpoint/uri**. In the URL, **uri** indicates the resource path, that is, the path for API access.

Constraints

Only KVM nodes can be created. Non-KVM nodes cannot be used after being created.

Calling Method

For details, see [Calling APIs](#).

URI

POST /api/v3/projects/{project_id}/clusters/{cluster_id}/nodes

Table 4-226 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Table 4-227 Query Parameters

Parameter	Mandatory	Type	Description
nodepoolScaleUp	No	String	Whether the request is delivered by the node pool. If the value is not NodepoolScaleUp , the number of pods in the corresponding node pool is automatically updated.

Request Parameters

Table 4-228 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Table 4-229 Request body parameters

Parameter	Mandatory	Type	Description
kind	Yes	String	API type. The value is fixed at Node and cannot be changed.
apiVersion	Yes	String	API version. The value is fixed at v3 and cannot be changed.
metadata	No	NodeMetadata object	Node metadata, which is a collection of attributes.
spec	Yes	NodeSpec object	Detailed description of the node. CCE creates or updates objects by defining or updating spec .

Table 4-230 NodeMetadata

Parameter	Mandatory	Type	Description
name	No	String	<p>Node name</p> <p>NOTE</p> <p>Enter 1 to 56 characters starting with a letter and not ending with a hyphen (-). Only lowercase letters, digits, and hyphens (-) are allowed.</p> <p>If name is left empty or is not specified, the node name is generated based on the default rule, which is <i>Cluster name-Random characters</i>. Only the first 36 characters are kept for a long cluster name.</p> <p>If the number of nodes (count) is greater than 1, some random characters are added to the end of the entered node name according to the default rule, which is <i>User-defined name-Random characters</i>. Only the first 50 characters in the custom name are kept, followed by some random characters.</p> <p>Names of nodes running in clusters v1.28.1, v1.27.3, v1.25.6, v1.23.11, v1.21.12 or later support periods (.), but this configuration is not recommended.</p>

Parameter	Mandatory	Type	Description
uid	No	String	Node ID, which is unique and automatically generated after the resource is created. A user-defined ID will not take effect.
labels	No	Map<String,String>	CCE node label (not the native Kubernetes label). Labels are used to select objects that meet certain criteria. A label is a key-value pair. Example: <pre>"labels": { "key": "value" }</pre>
annotations	No	Map<String,String>	CCE node annotations, in key-value pairs (not the native Kubernetes annotations). Example: <pre>"annotations": { "key1": "value1", "key2": "value2" }</pre> NOTE <ul style="list-style-type: none"> annotations are not used to identify or select objects. The metadata in annotations may be small or large, structured or unstructured, and may include characters that are not allowed in labels. This parameter is used only for query and cannot be input through a request. Entered data of this parameter is invalid.
creationTimestamp	No	String	Time when the object was created. The value is automatically generated after the object is created. A user-defined value will not take effect.
updateTimestamp	No	String	Time when the object was updated. The value is automatically generated after the object is created. A user-defined value will not take effect.

Table 4-231 NodeSpec

Parameter	Mandatory	Type	Description
flavor	Yes	String	Node specifications. For details about the node specifications supported by CCE, see Node Flavor Description .
az	Yes	String	AZ where the node to be created is located. You need to specify the AZ name. If it is set to random , the node will be created in a random AZ. For details about AZs supported by CCE, see Regions and Endpoints .
os	No	String	Node OS. For details about the supported OSs, see Node OS . NOTE <ul style="list-style-type: none"> The system automatically selects the supported OS based on the cluster version. If the current cluster version does not support the OS, an error will be reported. If alpha.cce/NodeImageID in extendParam is specified during node creation, you do not need to configure this parameter. This parameter is mandatory when creating a node pool. If you are using shared disk space when creating a node, this parameter is mandatory. Specifically, the disk initialization configuration management parameter storage is used and the name field of virtualSpaces in StorageGroups is set to share.
login	Yes	Login object	Node login mode. Either the key pair or password must be used for login.
rootVolume	Yes	Volume object	Information about disks on the node

Parameter	Mandatory	Type	Description
dataVolumes	Yes	Array of Volume objects	<p>Data disk parameters of the node. Currently, you can add the second data disk for your node on the CCE console.</p> <p>This data disk is used by the container runtime and kubelet. Do not uninstall this disk. Otherwise, the node will become unavailable.</p> <p>For DeC nodes, the parameter description is the same as that for rootVolume.</p>
storage	No	Storage object	<p>Disk initialization management parameter.</p> <p>This parameter is complex to configure. For details, see Attaching Disks to a Node.</p> <p>If this parameter retains its default, disks are managed based on the DockerLVMConfigOverride (discarded) parameter in extendParam. This parameter is supported by clusters of version 1.15.11 and later.</p> <p>NOTE</p> <ul style="list-style-type: none"> • If a node specification involves both local disks and EVS disks, do not retain the default value of this parameter to prevent unexpected disk partitions. • If you want to change the value range of a data disk to 20 to 32768, do not use the default value. • If you want to use the shared disk space (with the runtime and Kubernetes partitions cancelled), do not retain the default value of this parameter. For details about the shared disk space, see Data Disk Space Allocation.
publicIP	No	NodePublicIP object	<p>EIP of a node.</p> <p>NOTE</p> <p>This parameter is not supported when you add a node to a node pool.</p>

Parameter	Mandatory	Type	Description
nodeNicSpec	No	NodeNicSpec object	NIC of the node
count	No	Integer	Number of nodes to be created in a batch. The value must be a positive integer greater than or equal to 1 and less than or equal to the defined limit. This parameter can be left blank when it is used for a node pool.
billingMode	No	Integer	Node billing mode. <ul style="list-style-type: none"> • 0: pay-per-use • 1: yearly/monthly • 2: (discarded) yearly/monthly billing with auto payment enabled

Parameter	Mandatory	Type	Description
taints	No	Array of Taint objects	<p>You can add taints to created nodes to configure anti-affinity. A maximum of 20 taints can be added. Each taint contains the following parameters:</p> <ul style="list-style-type: none"> • Key: A key must contain 1 to 63 characters starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain name can be used as the prefix of a key. • Value: A value must start with a letter or digit and can contain a maximum of 63 characters, including letters, digits, hyphens (-), underscores (_), and periods (.). • Effect: Available options are NoSchedule, PreferNoSchedule, and NoExecute. When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query. <p>Example:</p> <pre>"taints": [{ "key": "status", "value": "unavailable", "effect": "NoSchedule" }, { "key": "looks", "value": "bad", "effect": "NoSchedule" }]</pre>

Parameter	Mandatory	Type	Description
k8sTags	No	Map<String,String>	<p>Defined in key-value pairs. A maximum of 20 key-value pairs are allowed.</p> <ul style="list-style-type: none"> • Key: Enter 1 to 63 characters, starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain can be prefixed to a key and contain a maximum of 253 characters. Example DNS subdomain: example.com/my-key • Value: The value can be left blank or contain 1 to 63 characters that start with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed in the character string. <p>When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query.</p> <p>Example:</p> <pre>"k8sTags": { "key": "value" }</pre>
ecsGroupId	No	String	<p>ECS group ID. If this parameter is configured, nodes will be created in the specified ECS group.</p> <p>NOTE This configuration does not take effect when a node pool is created. To ensure that all nodes in a node pool are in the same ECS group, configure nodeManagement in the node pool.</p>

Parameter	Mandatory	Type	Description
dedicatedHostId	No	String	<p>ID of the DeH to which the node is scheduled.</p> <p>NOTE This parameter is not supported when you add a node during node pool creation.</p>
userTags	No	Array of UserTag objects	<p>Cloud server tag. The key of a tag must be unique. The maximum number of custom tags supported by CCE depends on the region and cannot exceed 8.</p> <p>When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query.</p> <p>NOTE Use only letters, Unicode characters, digits, special characters(-, _). Max characters: 36</p>
runtime	No	Runtime object	<p>Container runtime:</p> <ul style="list-style-type: none"> • Clusters of v1.25 or earlier: docker. • Clusters of v1.25 or later: Container runtime varies with the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker. For nodes running other OSs, the default container runtime is containerd.

Parameter	Mandatory	Type	Description
initializedConditions	No	Array of strings	<p>Custom initialization flag, which is left blank by default. Before CCE nodes are initialized, node.cloudprovider.kubernetes.io/uninitialized is added to the nodes to prevent pods from being scheduled to these nodes. When creating a node, you can configure the initializedConditions parameter to control the taint removal time. By default, the timeout period is not configured.</p> <p>Example:</p> <ol style="list-style-type: none"> 1. Create a node and add parameter <code>"initializedConditions": ["CCEInitial", "CustomedInitial"]</code>. 2. After custom initialization is complete, CCE will call a Kubernetes API (for example, PATCH /v1/nodes/{node_ip}/status) to update the node conditions by adding two labels of types CCEInitial and CustomedInitial and setting the status to True, as shown in the following: <pre>status: conditions: - type: CCEInitial status: 'True' - type: CustomedInitial status: 'True'</pre> <ol style="list-style-type: none"> 1. CCE polls status.Conditions of nodes to check whether there are conditions of types CCEInitial and CustomedInitial. If such conditions exist and the status is True, the node initialization is complete and the initialization taint is removed.

Parameter	Mandatory	Type	Description
			<p>2. initializedConditions allows you to configure a timeout period during node creation. For example, "initializedConditions": ["CCEInitial:15m", "CustomedInitial:15m"] indicates that the timeout period is 15 minutes. After the timeout period is reached, the initialization conditions will be automatically ignored and the initialization taint will be removed when CCE polls the node.</p> <ul style="list-style-type: none"> • Use only letters and digits. Max. characters: 20. • Max. flags: 2. • The unit of the timeout period is minute (m).
extendParam	No	NodeExtendParam object	Extended parameters for creating a node.
hostnameConfig	No	HostnameConfig object	Kubernetes node name configuration parameter, which is supported by clusters of v1.23.6-r0 to v1.25 or clusters of v1.25.2-r0 or later versions.

Table 4-232 Login

Parameter	Mandatory	Type	Description
sshKey	No	String	Name of the key pair used for login.
userPassword	No	UserPassword object	Password used for node login.

Table 4-233 UserPassword

Parameter	Mandatory	Type	Description
username	No	String	Login account. The default value is root .
password	Yes	String	<p>If a username and a password are used to create a node, this field is shielded in the response body.</p> <p>A password must meet the following complexity requirements:</p> <ul style="list-style-type: none"> • Contains 8 to 26 characters. • Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (!@\$%^_-=+[{ }];,./?~#*) • Cannot contain the username or the username spelled backwards. <p>The password field must be salted during node creation. For details, see Adding a Salt in the password Field When Creating a Node.</p>

Table 4-234 Volume

Parameter	Mandatory	Type	Description
size	Yes	Integer	<p>Disk size, in GB.</p> <ul style="list-style-type: none"> • System disk: 40 to 1024 • Value range for the first data disk: 20 to 32768 (When storage is not specified, this value range becomes 100 to 32768.) • Value range for other data disks: 10 to 32768. (When storage is not specified, this value range becomes 100 to 32768.)

Parameter	Mandatory	Type	Description
volumetype	Yes	String	Disk type. For details about possible values, see the description of the root_volume parameter in the API used to create an ECS. <ul style="list-style-type: none"> • SAS: high I/O SAS disk • SSD: ultra-high I/O SSD disk • SATA: common I/O SATA disk SATA disks have been removed from EVS. You can find them attached only on existing nodes.
extendParam	No	Map<String,Object>	Extended disk parameters, defined in extendparam in the API used to create an ECS. For details, see Creating an ECS .
cluster_id	No	String	ID of the storage pool used by the ECS system disk. This field is used only for DeC clusters, which functions as dssPoolID , that is, the ID of the DSS storage pool. To obtain the value, see the ID field in the API used to obtain details of a DSS storage pool .
cluster_type	No	String	Storage class of the cloud server system disk. The value is always dss . This field is used only for DeC clusters.

Parameter	Mandatory	Type	Description
hw:passthrough	No	Boolean	<ul style="list-style-type: none"> Pay attention to this field if your ECS is SDI-compliant. If the value of this field is true, an SCSI disk will be created. If the node pool type is ElasticBMS, this field must be set to true. If a node specification involves local disks and EVS disks at the same time, set the disk initialization parameters. For details, see Attaching Disks to a Node.
metadata	No	VolumeMetadata object	EVS disk encryption information. This field is mandatory only when you need to encrypt the system disk or data disks of the node to be created.

Table 4-235 VolumeMetadata

Parameter	Mandatory	Type	Description
__system__encrypted	No	String	<p>Whether the EVS disk is encrypted. The value 0 indicates that the EVS disk is not encrypted, and the value 1 indicates that the EVS disk is encrypted.</p> <p>If this parameter is not specified, EVS disks will not be encrypted by default.</p>
__system__cmkid	No	String	CMK ID, which indicates encryption in metadata . This field is used with __system__encrypted .

Table 4-236 Storage

Parameter	Mandatory	Type	Description
storageSelectors	Yes	Array of StorageSelectors objects	Disk selection. Matched disks are managed according to matchLabels and storageType .
storageGroups	Yes	Array of StorageGroups objects	A storage group consists of multiple storage devices. It is used to divide storage space.

Table 4-237 StorageSelectors

Parameter	Mandatory	Type	Description
name	Yes	String	Selector name, used as the index of selectorNames in storageGroup . Therefore, the name of each selector must be unique.
storageType	Yes	String	Specifies the storage type. Currently, only evs (EVS volumes) and local (local volumes) are supported. The local storage does not support disk selection. All local disks will form a VG. Therefore, only one storageSelector of the local type is allowed.
matchLabels	No	matchLabels object	Matching field of an EVS volume. The size , volumeType , metadataEncrypted , metadataCmkid and count fields are supported.

Table 4-238 matchLabels

Parameter	Mandatory	Type	Description
size	No	String	Matched disk size. If this parameter is left unspecified, the disk size is not limited. Example: 100

Parameter	Mandatory	Type	Description
volumeType	No	String	EVS disk type. SSD, GPSSD, SAS, ESSD, and SATA are supported. If this parameter is left blank, there is no restriction on the disk type.
metadataEncrypted	No	String	Disk encryption identifier. If this parameter is set to 0 , it specifies that the disk is not encrypted. If this parameter is set to 1 , it specifies that the disk is encrypted. If this parameter is left blank, there is no restriction on the disk encryption identifier.
metadataCmkid	No	String	CMK ID of an encrypted disk. The value is a string of 36 bytes. If this parameter is left blank, there is no restriction on the disk key ID.
count	No	String	Number of disks to be selected. If this parameter is left blank, all disks of this type are selected.

Table 4-239 StorageGroups

Parameter	Mandatory	Type	Description
name	Yes	String	Name of a virtual storage group, which must be unique. NOTE <ul style="list-style-type: none"> If cceManaged is set to true, the name must be set to vgpaas. If the data disk is used as a temporary storage volume, the name must be vg-everest-localvolume-ephemeral. If the data disk is used as a persistent storage volume, the name must be vg-everest-localvolume-persistent.

Parameter	Mandatory	Type	Description
cceManaged	No	Boolean	Storage space for Kubernetes and runtime components. Only one group can be set to true . If this parameter is left blank, the default value false is used.
selectorNames	Yes	Array of strings	This parameter corresponds to name in storageSelectors . A group can match multiple selectors, but a selector can match only one group.
virtualSpaces	Yes	Array of VirtualSpace objects	Detailed management of space configuration in a group.

Table 4-240 VirtualSpace

Parameter	Mandatory	Type	Description
name	Yes	String	Name of a virtualSpace. Options: share , kubernetes , runtime , and user <ul style="list-style-type: none"> • share: shared space configuration (with the runtime and Kubernetes partitions cancelled). lvmConfig needs to be configured. • Kubernetes: Kubernetes space configuration. lvmConfig needs to be configured. • runtime: runtime space configuration. runtimeConfig needs to be configured. • user: user space configuration. lvmConfig needs to be configured.
size	Yes	String	Size of a virtualSpace. The value must be an integer in percentage. Example: 90%. NOTE The sum of the percentages of all virtualSpaces in a group cannot exceed 100%.

Parameter	Mandatory	Type	Description
lvmConfig	No	LVMConfig object	LVM configuration managements, applicable to the <i>share*</i> , kubernetes , and user spaces. Note that one virtual space supports only one config.
runtimeConfig	No	RuntimeConfig object	runtime configurations, applicable to the runtime space. Note that one virtual space supports only one config.

Table 4-241 LVMConfig

Parameter	Mandatory	Type	Description
lvType	Yes	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.
path	No	String	Path to which the disk is attached. This parameter takes effect only in user configuration. The value is an absolute path. Digits, letters, periods (.), hyphens (-), and underscores (_) are allowed.

Table 4-242 RuntimeConfig

Parameter	Mandatory	Type	Description
lvType	Yes	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.

Table 4-243 NodePublicIP

Parameter	Mandatory	Type	Description
ids	No	Array of strings	IDs of existing EIPs. The quantity cannot be greater than the number of nodes to be created. NOTE If ids has been set, you do not need to set count and eip .
count	No	Integer	Number of EIPs to be dynamically created. NOTE count and eip must be set at the same time.
eip	No	NodeEIPSpec object	EIP configuration.

Table 4-244 NodeEIPSpec

Parameter	Mandatory	Type	Description
iptype	Yes	String	EIP type, specified in publicip.type in the API for assigning an EIP. For details, see Assigning an EIP .
bandwidth	No	NodeBandwidth object	Bandwidth parameters of the EIP

Table 4-245 NodeBandwidth

Parameter	Mandatory	Type	Description
chargemode	No	String	<p>Bandwidth billing mode.</p> <ul style="list-style-type: none"> If this field is not specified, the billing is based on bandwidth. If the field is null, the billing is based on bandwidth. If the field value is traffic, the billing is based on traffic. If the value is out of the preceding options, the cloud server will fail to be created. <p>NOTE</p> <ul style="list-style-type: none"> Billed by bandwidth: The billing will be based on the data transmission rate (in Mbps) of public networks. This billing mode is recommended if your bandwidth usage is higher than 10%. Billed by traffic: The billing is based on the total amount of data (in GB) transmitted over the public network. This mode is available only when you are creating a pay-per-use node. This billing mode is recommended if your bandwidth usage is lower than 10%.
size	No	Integer	<p>Bandwidth size, specified in bandwidth.size in the API for assigning an EIP.</p> <p>For details, see Assigning an EIP.</p>
sharetype	No	String	<p>Bandwidth sharing type. Value options: PER (exclusive bandwidth)</p>

Table 4-246 NodeNicSpec

Parameter	Mandatory	Type	Description
primaryNic	No	NicSpec object	Description of the primary NIC.
extNics	No	Array of NicSpec objects	Extension NIC NOTE This parameter is not supported when you add a node to a node pool.

Table 4-247 NicSpec

Parameter	Mandatory	Type	Description
subnetId	No	String	Network ID of the subnet to which a NIC belongs. If subnetId is not specified when a primary NIC is creating, the cluster subnet will be used. If subnetList is also configured for a node pool, the subnetList field is used for adding subnets to the node pool. When creating an extension NIC, you must specify subnetId .
fixedIps	No	Array of strings	The IP address of the primary ENI is specified using fixedIps . The number of IP addresses cannot be greater than the number of created nodes. Either fixedIps or ipBlock can be specified. fixedIps cannot be specified for extension ENIs.
ipBlock	No	String	CIDR format of the primary NIC IP range. The IP address of the created node falls in this range. fixedIps and ipBlock cannot be specified at the same time.

Table 4-248 Taint

Parameter	Mandatory	Type	Description
key	Yes	String	Key.
value	No	String	Value.
effect	Yes	String	Effect.

Table 4-249 UserTag

Parameter	Mandatory	Type	Description
key	No	String	Key of the cloud server label. The value cannot start with CCE- or __type_baremetal .
value	No	String	Value of the cloud server label.

Table 4-250 Runtime

Parameter	Mandatory	Type	Description
name	No	String	Container runtime. Default value: <ul style="list-style-type: none"> Clusters earlier than v1.25: The default value is docker. Clusters of v1.25 or later: The default container runtime varies depending on the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker; for nodes running other OSs, the default container runtime is containerd.

Table 4-251 NodeExtendParam

Parameter	Mandatory	Type	Description
ecs:performancetype	No	String	ECS flavor types. This field is returned in the response.

Parameter	Mandatory	Type	Description
orderId	No	String	Order ID. This parameter is returned in the response when the node is billed on a yearly/ monthly basis with auto payment enabled (only in creation scenarios).
productID	No	String	Product ID. This field is returned in the response when the node is yearly/monthly-billed with auto payment enabled.
maxPods	No	Integer	Maximum number of pods that can be created on a node, including the default system pods. Value range: 16 to 256. This limit prevents the node from being overloaded of pods. The number of pods that can be created on a node is determined by multiple parameters. For details, see Maximum Number of Pods That Can Be Created on a Node .
periodType	No	String	<ul style="list-style-type: none"> ● month: The unit is month. ● year: The unit is year. <p>NOTE Request parameter, which is valid and mandatory only when billingMode is set to 1 (yearly/ monthly billing) or 2 (yearly/ monthly billing with auto renewal enabled) (discarded). Response parameter, which is returned only when a yearly/ monthly node is created.</p>

Parameter	Mandatory	Type	Description
periodNum	No	Integer	<p>Subscription duration. The value can be:</p> <ul style="list-style-type: none"> • If periodType is month, the value ranges from 1 to 9. • If periodType is year, the value is 1. <p>NOTE Request parameter, which is valid and mandatory when billingMode is set to 1 or 2 (discarded). Response parameter, which is returned only when a yearly/monthly node is created.</p>
isAutoRenew	No	String	<p>Whether auto renewal is enabled.</p> <ul style="list-style-type: none"> • true: Auto renewal is enabled. • false: Auto renewal is not enabled. <p>NOTE This field is valid when billingMode is set to 1 or 2 (discarded). If not specified, auto renewal is not enabled.</p>
isAutoPay	No	String	<p>Whether to deduct fees automatically.</p> <ul style="list-style-type: none"> • true: Enable automatic fee deduction. • false: Do not enable automatic fee deduction. <p>NOTE This field is valid when billingMode is set to 1 or 2 (discarded). If not specified when billingMode is set to 1, auto fee deduction is not enabled. (Discarded: If not specified when billingMode is set to 2, auto fee deduction is enabled.)</p>

Parameter	Mandatory	Type	Description
DockerLVMConfigOverride	No	String	<p>Docker data disk configuration item. (This parameter has been discarded. Use the storage field instead.)</p> <p>Example default configuration: <code>"DockerLVMConfigOverride":"dockerThinpool=vgpaas/90%VG;kubernetesLV=vgpaas/10%VG;diskType=evs;lvType=linear"</code></p> <p>By default, if no VD disk is available, an error occurs because the data disk fails to be found. Set diskType based on the actual drive letter type.</p> <p>The following fields are included:</p> <ul style="list-style-type: none"> • userLV (optional): size of the user space, for example, vgpaas/20%VG. • userPath (optional): mount path of the user space, for example, /home/wqt-test. • diskType: disk type. Currently, only evs, hdd, and ssd are supported. • lvType: type of a logic volume. The value can be linear or striped. • dockerThinpool: Docker space size, for example, vgpaas/60%VG. • kubernetesLV: kubelet space size, for example, vgpaas/20%VG.

Parameter	Mandatory	Type	Description
dockerBaseSize	No	Integer	<p>Available disk space of a single container on a node, in GB. This parameter has been discarded. Use containerBaseSize.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In Device Mapper mode, the default value is 10. In OverlayFS mode, the available space of a single container is not limited by default, and the dockerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters of the new versions.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that dockerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device.</p>

Parameter	Mandatory	Type	Description
containerBaseSize	No	Integer	<p>Available disk space of a single container on a node, in GB.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In OverlayFS mode, the available space of a single container is not limited by default. In Device Mapper mode, the default value is 10, and the containerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters v1.23.14-r0, v1.25.9-r0, v1.27.6-r0, v1.28.4-r0 or later.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that containerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device. In new versions, the Device Mapper mode is used only by BMSs in the same resource pool and is being discarded.</p>
publicKey	No	String	Public key of a node.
alpha.cce/preInstall	No	String	<p>Pre-installation script.</p> <p>NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)</p>

Parameter	Mandatory	Type	Description
alpha.cce/postInstall	No	String	Post-installation script. NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)
alpha.cce/NodeImageID	No	String	This parameter is required when a custom image is used to create a BMS node.
nicMultiqueue	No	String	<ul style="list-style-type: none"> Number of ENI queues. Example setting: <code>"[{\"queue\":4}]"</code> The following fields are included: <ul style="list-style-type: none"> queue: number of ENI queues. This field can be configured only for BMS nodes in a CCE Turbo cluster. Supported proportions are {"1":128, "2":92, "4":92, "8":32, "16":16, "28":9}. That is, if there is one queue, a maximum of 128 ENIs can be bound. If there are two queues, a maximum of 92 ENIs can be bound for two queues. A larger number of ENI queues indicates higher performance but fewer ENIs can be bound. The queue settings cannot be changed after the node pool is created.

Parameter	Mandatory	Type	Description
nicThreshold	No	String	<ul style="list-style-type: none"> ENI pre-binding thresholds. Example setting: "0.3:0.6" Low threshold (L): determines the minimum number of pre-bound ENIs (Min). Formula: $Min = Total\ number\ of\ ENIs\ of\ the\ node \times L$ High threshold (H): determines the maximum number of pre-bound ENIs (Max). Formula: $Max = Total\ number\ of\ ENIs\ of\ the\ node \times H$ Number of ENIs bound to a BMS node (B) and number of ENIs being used by pods (U): $U + Min < B < U + Max$ If the number of pre-bound ENIs on a BMS node is smaller than the minimum allowed, the system will bind more ENIs to make the numbers equal. If the number of pre-bound ENIs on a BMS node is larger than the maximum allowed, the system periodically unbinds ENIs (about every 2 minutes) to make the numbers equal. Both the thresholds are one-decimal-place values ranging from 0.0 to 1.0. The low threshold must be smaller than or equal to the high one. This field can be configured only for BMS nodes in a CCE Turbo cluster. Pre-binding ENIs can speed up workload creation but occupies IP addresses.

Parameter	Mandatory	Type	Description
chargingMode	No	Integer	Billing mode of a node. This parameter has been deprecated. Use the billingMode parameter in NodeSpec .
agency_name	No	String	Name of an agency An agency is created by a tenant administrator on Identity and Access Management (IAM) to provide temporary credentials for CCE nodes to access cloud servers. This parameter is returned only when it is transferred during node creation.
kubeReservedMem	No	Integer	Reserved node memory, which is reserved for Kubernetes components.
systemReservedMem	No	Integer	Reserved node memory, which is reserved for system components.
init-node-password	No	String	Node password. If this parameter is used as a response parameter, asterisks (*) will be displayed.

Table 4-252 HostnameConfig

Parameter	Mandatory	Type	Description
type	Yes	String	<p>Configuration type of the Kubernetes node name. The default value is privatelp.</p> <ul style="list-style-type: none"> • privatelp: The Kubernetes node is named after its IP address. • cceNodeName: The Kubernetes node is named after the CCE node. <p>NOTE</p> <ul style="list-style-type: none"> • For a node which is configured using cceNodeName, the name is the same as the Kubernetes node name and the ECS name. The node name cannot be changed. If the ECS name is changed on the ECS console, the node name will retain unchanged after ECS synchronization. • For a node which is configured using cceNodeName, to avoid a conflict between Kubernetes nodes, the system automatically adds a suffix to each node name. The suffix is in the format of A hyphen (-) Five random characters. The value of the random characters is a lowercase letter or a digit ranging from 0 to 9.

Response Parameters

Status code: 201

Table 4-253 Response body parameters

Parameter	Type	Description
kind	String	API type. The value is fixed at Node and cannot be changed.
apiVersion	String	API version. The value is fixed at v3 and cannot be changed.

Parameter	Type	Description
metadata	NodeMetadata object	Node metadata, which is a collection of attributes.
spec	NodeSpec object	Detailed description of the node. CCE creates or updates objects by defining or updating spec .
status	NodeStatus object	Node status, which is dynamically recorded. A user-defined value will not function when a node is being created or modified.

Table 4-254 NodeMetadata

Parameter	Type	Description
name	String	<p>Node name</p> <p>NOTE Enter 1 to 56 characters starting with a letter and not ending with a hyphen (-). Only lowercase letters, digits, and hyphens (-) are allowed.</p> <p>If name is left empty or is not specified, the node name is generated based on the default rule, which is <i>Cluster name-Random characters</i>. Only the first 36 characters are kept for a long cluster name.</p> <p>If the number of nodes (count) is greater than 1, some random characters are added to the end of the entered node name according to the default rule, which is <i>User-defined name-Random characters</i>. Only the first 50 characters in the custom name are kept, followed by some random characters.</p> <p>Names of nodes running in clusters v1.28.1, v1.27.3, v1.25.6, v1.23.11, v1.21.12 or later support periods (.), but this configuration is not recommended.</p>
uid	String	Node ID, which is unique and automatically generated after the resource is created. A user-defined ID will not take effect.
labels	Map<String,String>	<p>CCE node label (not the native Kubernetes label).</p> <p>Labels are used to select objects that meet certain criteria. A label is a key-value pair.</p> <p>Example:</p> <pre>"labels": { "key": "value" }</pre>

Parameter	Type	Description
annotations	Map<String,String>	<p>CCE node annotations, in key-value pairs (not the native Kubernetes annotations).</p> <p>Example:</p> <pre>"annotations": { "key1" : "value1", "key2" : "value2" }</pre> <p>NOTE</p> <ul style="list-style-type: none"> • annotations are not used to identify or select objects. The metadata in annotations may be small or large, structured or unstructured, and may include characters that are not allowed in labels. • This parameter is used only for query and cannot be input through a request. Entered data of this parameter is invalid.
creationTimestamp	String	Time when the object was created. The value is automatically generated after the object is created. A user-defined value will not take effect.
updateTimestamp	String	Time when the object was updated. The value is automatically generated after the object is created. A user-defined value will not take effect.

Table 4-255 NodeSpec

Parameter	Type	Description
flavor	String	Node specifications. For details about the node specifications supported by CCE, see Node Flavor Description .
az	String	<p>AZ where the node to be created is located. You need to specify the AZ name. If it is set to random, the node will be created in a random AZ.</p> <p>For details about AZs supported by CCE, see Regions and Endpoints.</p>

Parameter	Type	Description
os	String	<p>Node OS. For details about the supported OSs, see Node OS.</p> <p>NOTE</p> <ul style="list-style-type: none"> The system automatically selects the supported OS based on the cluster version. If the current cluster version does not support the OS, an error will be reported. If <code>alpha.cce/NodeImageID</code> in <code>extendParam</code> is specified during node creation, you do not need to configure this parameter. This parameter is mandatory when creating a node pool. If you are using shared disk space when creating a node, this parameter is mandatory. Specifically, the disk initialization configuration management parameter <code>storage</code> is used and the <code>name</code> field of <code>virtualSpaces</code> in <code>StorageGroups</code> is set to <code>share</code>.
login	Login object	Node login mode. Either the key pair or password must be used for login.
rootVolume	Volume object	Information about disks on the node
dataVolumes	Array of Volume objects	<p>Data disk parameters of the node. Currently, you can add the second data disk for your node on the CCE console.</p> <p>This data disk is used by the container runtime and kubelet. Do not uninstall this disk. Otherwise, the node will become unavailable.</p> <p>For DeC nodes, the parameter description is the same as that for <code>rootVolume</code>.</p>

Parameter	Type	Description
storage	Storage object	<p>Disk initialization management parameter. This parameter is complex to configure. For details, see Attaching Disks to a Node.</p> <p>If this parameter retains its default, disks are managed based on the DockerLVMConfigOverride (discarded) parameter in extendParam. This parameter is supported by clusters of version 1.15.11 and later.</p> <p>NOTE</p> <ul style="list-style-type: none"> • If a node specification involves both local disks and EVS disks, do not retain the default value of this parameter to prevent unexpected disk partitions. • If you want to change the value range of a data disk to 20 to 32768, do not use the default value. • If you want to use the shared disk space (with the runtime and Kubernetes partitions cancelled), do not retain the default value of this parameter. For details about the shared disk space, see Data Disk Space Allocation.
publicIP	NodePublicIP object	<p>EIP of a node.</p> <p>NOTE This parameter is not supported when you add a node to a node pool.</p>
nodeNicSpec	NodeNicSpec object	NIC of the node
count	Integer	Number of nodes to be created in a batch. The value must be a positive integer greater than or equal to 1 and less than or equal to the defined limit. This parameter can be left blank when it is used for a node pool.
billingMode	Integer	<p>Node billing mode.</p> <ul style="list-style-type: none"> • 0: pay-per-use • 1: yearly/monthly • 2: (discarded) yearly/monthly billing with auto payment enabled

Parameter	Type	Description
taints	Array of Taint objects	<p>You can add taints to created nodes to configure anti-affinity. A maximum of 20 taints can be added. Each taint contains the following parameters:</p> <ul style="list-style-type: none"> • Key: A key must contain 1 to 63 characters starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain name can be used as the prefix of a key. • Value: A value must start with a letter or digit and can contain a maximum of 63 characters, including letters, digits, hyphens (-), underscores (_), and periods (.). • Effect: Available options are NoSchedule, PreferNoSchedule, and NoExecute. When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query. <p>Example:</p> <pre>"taints": [{ "key": "status", "value": "unavailable", "effect": "NoSchedule" }, { "key": "looks", "value": "bad", "effect": "NoSchedule" }]</pre>

Parameter	Type	Description
k8sTags	Map<String,String>	<p>Defined in key-value pairs. A maximum of 20 key-value pairs are allowed.</p> <ul style="list-style-type: none"> Key: Enter 1 to 63 characters, starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain can be prefixed to a key and contain a maximum of 253 characters. Example DNS subdomain: example.com/my-key Value: The value can be left blank or contain 1 to 63 characters that start with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed in the character string. When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query. <p>Example:</p> <pre>"k8sTags": { "key": "value" }</pre>
ecsGroupId	String	<p>ECS group ID. If this parameter is configured, nodes will be created in the specified ECS group.</p> <p>NOTE This configuration does not take effect when a node pool is created. To ensure that all nodes in a node pool are in the same ECS group, configure nodeManagement in the node pool.</p>
dedicatedHostId	String	<p>ID of the DeH to which the node is scheduled.</p> <p>NOTE This parameter is not supported when you add a node during node pool creation.</p>

Parameter	Type	Description
userTags	Array of UserTag objects	<p>Cloud server tag. The key of a tag must be unique. The maximum number of custom tags supported by CCE depends on the region and cannot exceed 8.</p> <p>When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query.</p> <p>NOTE Use only letters, Unicode characters, digits, special characters(-, _). Max characters: 36</p>
runtime	Runtime object	<p>Container runtime:</p> <ul style="list-style-type: none"> • Clusters of v1.25 or earlier: docker. • Clusters of v1.25 or later: Container runtime varies with the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker. For nodes running other OSs, the default container runtime is containerd.

Parameter	Type	Description
initializedConditions	Array of strings	<p>Custom initialization flag, which is left blank by default.</p> <p>Before CCE nodes are initialized, node.cloudprovider.kubernetes.io/uninitialized is added to the nodes to prevent pods from being scheduled to these nodes. When creating a node, you can configure the initializedConditions parameter to control the taint removal time. By default, the timeout period is not configured.</p> <p>Example:</p> <ol style="list-style-type: none"> 1. Create a node and add parameter "initializedConditions": ["CCEInitial", "CustomedInitial"]. 2. After custom initialization is complete, CCE will call a Kubernetes API (for example, PATCH /v1/nodes/{node_ip}/status) to update the node conditions by adding two labels of types CCEInitial and CustomedInitial and setting the status to True, as shown in the following: <pre data-bbox="817 1055 1430 1205"> status: conditions: - type: CCEInitial status: 'True' - type: CustomedInitial status: 'True' </pre> <ol style="list-style-type: none"> 1. CCE polls status.Conditions of nodes to check whether there are conditions of types CCEInitial and CustomedInitial. If such conditions exist and the status is True, the node initialization is complete and the initialization taint is removed. 2. initializedConditions allows you to configure a timeout period during node creation. For example, "initializedConditions": ["CCEInitial:15m", "CustomedInitial:15m"] indicates that the timeout period is 15 minutes. After the timeout period is reached, the initialization conditions will be automatically ignored and the initialization taint will be removed when CCE polls the node. <ul style="list-style-type: none"> ● Use only letters and digits. Max. characters: 20. ● Max. flags: 2. ● The unit of the timeout period is minute (m).

Parameter	Type	Description
extendParam	NodeExtendParam object	Extended parameters for creating a node.
hostnameConfig	HostnameConfig object	Kubernetes node name configuration parameter, which is supported by clusters of v1.23.6-r0 to v1.25 or clusters of v1.25.2-r0 or later versions.

Table 4-256 Login

Parameter	Type	Description
sshKey	String	Name of the key pair used for login.
userPassword	UserPassword object	Password used for node login.

Table 4-257 UserPassword

Parameter	Type	Description
username	String	Login account. The default value is root .
password	String	<p>If a username and a password are used to create a node, this field is shielded in the response body.</p> <p>A password must meet the following complexity requirements:</p> <ul style="list-style-type: none"> • Contains 8 to 26 characters. • Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (!@\$%^-_=+[{ } ; , / ? ~ # *) • Cannot contain the username or the username spelled backwards. <p>The password field must be salted during node creation. For details, see Adding a Salt in the password Field When Creating a Node.</p>

Table 4-258 Volume

Parameter	Type	Description
size	Integer	Disk size, in GB. <ul style="list-style-type: none"> System disk: 40 to 1024 Value range for the first data disk: 20 to 32768 (When storage is not specified, this value range becomes 100 to 32768.) Value range for other data disks: 10 to 32768. (When storage is not specified, this value range becomes 100 to 32768.)
volumetype	String	Disk type. For details about possible values, see the description of the root_volume parameter in the API used to create an ECS. <ul style="list-style-type: none"> SAS: high I/O SAS disk SSD: ultra-high I/O SSD disk SATA: common I/O SATA disk SATA disks have been removed from EVS. You can find them attached only on existing nodes.
extendParam	Map<String, Object>	Extended disk parameters, defined in extendparam in the API used to create an ECS. For details, see Creating an ECS .
cluster_id	String	ID of the storage pool used by the ECS system disk. This field is used only for DeC clusters, which functions as dssPoolID , that is, the ID of the DSS storage pool. To obtain the value, see the ID field in the API used to obtain details of a DSS storage pool .
cluster_type	String	Storage class of the cloud server system disk. The value is always dss . This field is used only for DeC clusters.
hw:passthrough	Boolean	<ul style="list-style-type: none"> Pay attention to this field if your ECS is SDI-compliant. If the value of this field is true, an SCSI disk will be created. If the node pool type is ElasticBMS, this field must be set to true. If a node specification involves local disks and EVS disks at the same time, set the disk initialization parameters. For details, see Attaching Disks to a Node.
metadata	VolumeMetadata object	EVS disk encryption information. This field is mandatory only when you need to encrypt the system disk or data disks of the node to be created.

Table 4-259 VolumeMetadata

Parameter	Type	Description
<code>__system__encrypted</code>	String	Whether the EVS disk is encrypted. The value 0 indicates that the EVS disk is not encrypted, and the value 1 indicates that the EVS disk is encrypted. If this parameter is not specified, EVS disks will not be encrypted by default.
<code>__system__cmkid</code>	String	CMK ID, which indicates encryption in metadata . This field is used with __system__encrypted .

Table 4-260 Storage

Parameter	Type	Description
<code>storageSelectors</code>	Array of StorageSelectors objects	Disk selection. Matched disks are managed according to matchLabels and storageType .
<code>storageGroups</code>	Array of StorageGroups objects	A storage group consists of multiple storage devices. It is used to divide storage space.

Table 4-261 StorageSelectors

Parameter	Type	Description
<code>name</code>	String	Selector name, used as the index of selectorNames in storageGroup . Therefore, the name of each selector must be unique.
<code>storageType</code>	String	Specifies the storage type. Currently, only evs (EVS volumes) and local (local volumes) are supported. The local storage does not support disk selection. All local disks will form a VG. Therefore, only one storageSelector of the local type is allowed.
<code>matchLabels</code>	matchLabels object	Matching field of an EVS volume. The size , volumeType , metadataEncrypted , metadataCmkid and count fields are supported.

Table 4-262 matchLabels

Parameter	Type	Description
size	String	Matched disk size. If this parameter is left unspecified, the disk size is not limited. Example: 100
volumeType	String	EVS disk type. SSD, GPSSD, SAS, ESSD, and SATA are supported. If this parameter is left blank, there is no restriction on the disk type.
metadataEncrypted	String	Disk encryption identifier. If this parameter is set to 0 , it specifies that the disk is not encrypted. If this parameter is set to 1 , it specifies that the disk is encrypted. If this parameter is left blank, there is no restriction on the disk encryption identifier.
metadataCmkid	String	CMK ID of an encrypted disk. The value is a string of 36 bytes. If this parameter is left blank, there is no restriction on the disk key ID.
count	String	Number of disks to be selected. If this parameter is left blank, all disks of this type are selected.

Table 4-263 StorageGroups

Parameter	Type	Description
name	String	Name of a virtual storage group, which must be unique. NOTE <ul style="list-style-type: none"> If cceManaged is set to true, the name must be set to vgpaas. If the data disk is used as a temporary storage volume, the name must be vg-everest-localvolume-ephemeral. If the data disk is used as a persistent storage volume, the name must be vg-everest-localvolume-persistent.
cceManaged	Boolean	Storage space for Kubernetes and runtime components. Only one group can be set to true . If this parameter is left blank, the default value false is used.
selectorNames	Array of strings	This parameter corresponds to name in storageSelectors . A group can match multiple selectors, but a selector can match only one group.

Parameter	Type	Description
virtualSpaces	Array of VirtualSpace objects	Detailed management of space configuration in a group.

Table 4-264 VirtualSpace

Parameter	Type	Description
name	String	Name of a virtualSpace. Options: share , kubernetes , runtime , and user <ul style="list-style-type: none"> • share: shared space configuration (with the runtime and Kubernetes partitions cancelled). lvmConfig needs to be configured. • Kubernetes: Kubernetes space configuration. lvmConfig needs to be configured. • runtime: runtime space configuration. runtimeConfig needs to be configured. • user: user space configuration. lvmConfig needs to be configured.
size	String	Size of a virtualSpace. The value must be an integer in percentage. Example: 90%. NOTE The sum of the percentages of all virtualSpaces in a group cannot exceed 100%.
lvmConfig	LVMConfig object	LVM configuration managements, applicable to the <i>share</i> *, kubernetes , and user spaces. Note that one virtual space supports only one config.
runtimeConfig	RuntimeConfig object	runtime configurations, applicable to the runtime space. Note that one virtual space supports only one config.

Table 4-265 LVMConfig

Parameter	Type	Description
lvType	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.

Parameter	Type	Description
path	String	Path to which the disk is attached. This parameter takes effect only in user configuration. The value is an absolute path. Digits, letters, periods (.), hyphens (-), and underscores (_) are allowed.

Table 4-266 RuntimeConfig

Parameter	Type	Description
lvType	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.

Table 4-267 NodePublicIP

Parameter	Type	Description
ids	Array of strings	IDs of existing EIPs. The quantity cannot be greater than the number of nodes to be created. NOTE If ids has been set, you do not need to set count and eip .
count	Integer	Number of EIPs to be dynamically created. NOTE count and eip must be set at the same time.
eip	NodeEIPSpec object	EIP configuration.

Table 4-268 NodeEIPSpec

Parameter	Type	Description
iptype	String	EIP type, specified in publicip.type in the API for assigning an EIP. For details, see Assigning an EIP .
bandwidth	NodeBandwidth object	Bandwidth parameters of the EIP

Table 4-269 NodeBandwidth

Parameter	Type	Description
chargemode	String	<p>Bandwidth billing mode.</p> <ul style="list-style-type: none"> If this field is not specified, the billing is based on bandwidth. If the field is null, the billing is based on bandwidth. If the field value is traffic, the billing is based on traffic. If the value is out of the preceding options, the cloud server will fail to be created. <p>NOTE</p> <ul style="list-style-type: none"> Billed by bandwidth: The billing will be based on the data transmission rate (in Mbps) of public networks. This billing mode is recommended if your bandwidth usage is higher than 10%. Billed by traffic: The billing is based on the total amount of data (in GB) transmitted over the public network. This mode is available only when you are creating a pay-per-use node. This billing mode is recommended if your bandwidth usage is lower than 10%.
size	Integer	<p>Bandwidth size, specified in bandwidth.size in the API for assigning an EIP.</p> <p>For details, see Assigning an EIP.</p>
sharetype	String	<p>Bandwidth sharing type. Value options: PER (exclusive bandwidth)</p>

Table 4-270 NodeNicSpec

Parameter	Type	Description
primaryNic	NicSpec object	Description of the primary NIC.
extNics	Array of NicSpec objects	<p>Extension NIC</p> <p>NOTE This parameter is not supported when you add a node to a node pool.</p>

Table 4-271 NicSpec

Parameter	Type	Description
subnetId	String	Network ID of the subnet to which a NIC belongs. If subnetId is not specified when a primary NIC is creating, the cluster subnet will be used. If subnetList is also configured for a node pool, the subnetList field is used for adding subnets to the node pool. When creating an extension NIC, you must specify subnetId .
fixedIps	Array of strings	The IP address of the primary ENI is specified using fixedIps . The number of IP addresses cannot be greater than the number of created nodes. Either fixedIps or ipBlock can be specified. fixedIps cannot be specified for extension ENIs.
ipBlock	String	CIDR format of the primary NIC IP range. The IP address of the created node falls in this range. fixedIps and ipBlock cannot be specified at the same time.

Table 4-272 Taint

Parameter	Type	Description
key	String	Key.
value	String	Value.
effect	String	Effect.

Table 4-273 UserTag

Parameter	Type	Description
key	String	Key of the cloud server label. The value cannot start with CCE- or __type_baremetal .
value	String	Value of the cloud server label.

Table 4-274 Runtime

Parameter	Type	Description
name	String	<p>Container runtime. Default value:</p> <ul style="list-style-type: none"> • Clusters earlier than v1.25: The default value is docker. • Clusters of v1.25 or later: The default container runtime varies depending on the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker; for nodes running other OSs, the default container runtime is containerd.

Table 4-275 NodeExtendParam

Parameter	Type	Description
ecs:performancetype	String	ECS flavor types. This field is returned in the response.
orderId	String	Order ID. This parameter is returned in the response when the node is billed on a yearly/monthly basis with auto payment enabled (only in creation scenarios).
productId	String	Product ID. This field is returned in the response when the node is yearly/monthly-billed with auto payment enabled.
maxPods	Integer	<p>Maximum number of pods that can be created on a node, including the default system pods. Value range: 16 to 256.</p> <p>This limit prevents the node from being overloaded of pods.</p> <p>The number of pods that can be created on a node is determined by multiple parameters. For details, see Maximum Number of Pods That Can Be Created on a Node.</p>
periodType	String	<ul style="list-style-type: none"> • month: The unit is month. • year: The unit is year. <p>NOTE</p> <p>Request parameter, which is valid and mandatory only when billingMode is set to 1 (yearly/monthly billing) or 2 (yearly/monthly billing with auto renewal enabled) (discarded).</p> <p>Response parameter, which is returned only when a yearly/monthly node is created.</p>

Parameter	Type	Description
periodNum	Integer	<p>Subscription duration. The value can be:</p> <ul style="list-style-type: none"> • If periodType is month, the value ranges from 1 to 9. • If periodType is year, the value is 1. <p>NOTE Request parameter, which is valid and mandatory when billingMode is set to 1 or 2 (discarded). Response parameter, which is returned only when a yearly/monthly node is created.</p>
isAutoRenew	String	<p>Whether auto renewal is enabled.</p> <ul style="list-style-type: none"> • true: Auto renewal is enabled. • false: Auto renewal is not enabled. <p>NOTE This field is valid when billingMode is set to 1 or 2 (discarded). If not specified, auto renewal is not enabled.</p>
isAutoPay	String	<p>Whether to deduct fees automatically.</p> <ul style="list-style-type: none"> • true: Enable automatic fee deduction. • false: Do not enable automatic fee deduction. <p>NOTE This field is valid when billingMode is set to 1 or 2 (discarded). If not specified when billingMode is set to 1, auto fee deduction is not enabled. (Discarded: If not specified when billingMode is set to 2, auto fee deduction is enabled.)</p>

Parameter	Type	Description
DockerLVMConfigOverride	String	<p>Docker data disk configuration item. (This parameter has been discarded. Use the storage field instead.) Example default configuration: "DockerLVMConfigOverride":"dockerThinpool=vgpaas/90%VG;kubernetesLV=vgpaas/10%VG;diskType=evs;lvType=linear"</p> <p>By default, if no VD disk is available, an error occurs because the data disk fails to be found. Set diskType based on the actual drive letter type.</p> <p>The following fields are included:</p> <ul style="list-style-type: none"> • userLV (optional): size of the user space, for example, vgpaas/20%VG. • userPath (optional): mount path of the user space, for example, /home/wqt-test. • diskType: disk type. Currently, only evs, hdd, and ssd are supported. • lvType: type of a logic volume. The value can be linear or striped. • dockerThinpool: Docker space size, for example, vgpaas/60%VG. • kubernetesLV: kubelet space size, for example, vgpaas/20%VG.
dockerBaseSize	Integer	<p>Available disk space of a single container on a node, in GB. This parameter has been discarded. Use containerBaseSize.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In Device Mapper mode, the default value is 10. In OverlayFS mode, the available space of a single container is not limited by default, and the dockerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters of the new versions.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that dockerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device.</p>

Parameter	Type	Description
containerBaseSize	Integer	<p>Available disk space of a single container on a node, in GB.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In OverlayFS mode, the available space of a single container is not limited by default. In Device Mapper mode, the default value is 10, and the containerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters v1.23.14-r0, v1.25.9-r0, v1.27.6-r0, v1.28.4-r0 or later.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that containerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device. In new versions, the Device Mapper mode is used only by BMSs in the same resource pool and is being discarded.</p>
publicKey	String	Public key of a node.
alpha.cce/preInstall	String	<p>Pre-installation script.</p> <p>NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)</p>
alpha.cce/postInstall	String	<p>Post-installation script.</p> <p>NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)</p>
alpha.cce/NodeImageID	String	This parameter is required when a custom image is used to create a BMS node.

Parameter	Type	Description
nicMultiqueue	String	<ul style="list-style-type: none"> Number of ENI queues. Example setting: <code>"[{"queue":4}]"</code> The following fields are included: <ul style="list-style-type: none"> queue: number of ENI queues. This field can be configured only for BMS nodes in a CCE Turbo cluster. Supported proportions are {"1":128, "2":92, "4":92, "8":32, "16":16, "28":9}. That is, if there is one queue, a maximum of 128 ENIs can be bound. If there are two queues, a maximum of 92 ENIs can be bound for two queues. A larger number of ENI queues indicates higher performance but fewer ENIs can be bound. The queue settings cannot be changed after the node pool is created.
nicThreshold	String	<ul style="list-style-type: none"> ENI pre-binding thresholds. Example setting: <code>"0.3:0.6"</code> <ul style="list-style-type: none"> Low threshold (L): determines the minimum number of pre-bound ENIs (Min). Formula: $Min = Total\ number\ of\ ENIs\ of\ the\ node \times L$ High threshold (H): determines the maximum number of pre-bound ENIs (Max). Formula: $Max = Total\ number\ of\ ENIs\ of\ the\ node \times H$ Number of ENIs bound to a BMS node (B) and number of ENIs being used by pods (U): $U + Min < B < U + Max$ If the number of pre-bound ENIs on a BMS node is smaller than the minimum allowed, the system will bind more ENIs to make the numbers equal. If the number of pre-bound ENIs on a BMS node is larger than the maximum allowed, the system periodically unbinds ENIs (about every 2 minutes) to make the numbers equal. Both the thresholds are one-decimal-place values ranging from 0.0 to 1.0. The low threshold must be smaller than or equal to the high one. This field can be configured only for BMS nodes in a CCE Turbo cluster. Pre-binding ENIs can speed up workload creation but occupies IP addresses.

Parameter	Type	Description
chargingMode	Integer	Billing mode of a node. This parameter has been deprecated. Use the billingMode parameter in NodeSpec .
agency_name	String	Name of an agency An agency is created by a tenant administrator on Identity and Access Management (IAM) to provide temporary credentials for CCE nodes to access cloud servers. This parameter is returned only when it is transferred during node creation.
kubeReservedMem	Integer	Reserved node memory, which is reserved for Kubernetes components.
systemReservedMem	Integer	Reserved node memory, which is reserved for system components.
init-node-password	String	Node password. If this parameter is used as a response parameter, asterisks (*) will be displayed.

Table 4-276 HostnameConfig

Parameter	Type	Description
type	String	Configuration type of the Kubernetes node name. The default value is privatelp . <ul style="list-style-type: none"> • privatelp: The Kubernetes node is named after its IP address. • cceNodeName: The Kubernetes node is named after the CCE node. <p>NOTE</p> <ul style="list-style-type: none"> • For a node which is configured using cceNodeName, the name is the same as the Kubernetes node name and the ECS name. The node name cannot be changed. If the ECS name is changed on the ECS console, the node name will retain unchanged after ECS synchronization. • For a node which is configured using cceNodeName, to avoid a conflict between Kubernetes nodes, the system automatically adds a suffix to each node name. The suffix is in the format of A hyphen (-) Five random characters. The value of the random characters is a lowercase letter or a digit ranging from 0 to 9.

Table 4-277 NodeStatus

Parameter	Type	Description
phase	String	Node status, which is the status of a node during its lifecycle (such as installation and uninstallation) or the status of a node in a Kubernetes cluster. Options: <ul style="list-style-type: none"> ● Build: The node is being created. ● Installing: The node is being managed. ● Upgrading: The node is being upgraded. ● Active: The node is running properly. ● Abnormal: The node is abnormal. ● Deleting: The node is being deleted. ● Error: The node is faulty.
lastProbeTime	String	Last time when the node status was checked. If the cluster is in the abnormal, frozen, or intermediate state (for example, creating), the node status check may be affected. The node status that takes more than five minutes to check has no reference value.
jobID	String	ID of a creation or deletion job
serverId	String	ID of the underlying ECS or BMS node
privateIP	String	IP address in the private network segment of the primary NIC on the node
privateIPv6IP	String	IPv6 address in the private network segment of the primary NIC on the node
publicIP	String	Node EIP. If the ECS data is not synchronized in real time, you can click Sync Node Data on the console to manually update the data.
deleteStatus	DeleteStatus object	Resource status during resource deletion.

Table 4-278 DeleteStatus

Parameter	Type	Description
previous_total	Integer	Total number of existing cluster resource records when the cluster is deleted.
current_total	Integer	Latest number of resource records, which is generated based on the current cluster resource records.

Parameter	Type	Description
updated	Integer	Total number of resource records updated when the cluster is deleted.
added	Integer	Total number of resource records updated when the cluster is deleted.
deleted	Integer	Total number of resource records deleted when the cluster is deleted.

Example Requests

- Create a yearly/monthly node with the flavor of 2 vCPUs and 4 GiB of memory. The node runs EulerOS 2.5 for Docker containers. The sizes of the system disk and data disk of the node are 50 GB and 100 GB, respectively. Both the disks are of the high I/O type.

POST /api/v3/projects/{project_id}/clusters/{cluster_id}/nodes

```
{
  "kind": "Node",
  "apiVersion": "v3",
  "metadata": {
    "name": "test-67235"
  },
  "spec": {
    "flavor": "c7.large.2",
    "az": "*****",
    "os": "EulerOS 2.5",
    "dataVolumes": [ {
      "size": 100,
      "volumetype": "SAS"
    } ],
    "billingMode": 1,
    "extendParam": {
      "maxPods": 110,
      "periodType": "month",
      "periodNum": 1,
      "isAutoPay": "false",
      "isAutoRenew": "false"
    },
    "nodeNicSpec": {
      "primaryNic": {
        "subnetId": "ca964acf-8468-4735-8229-97940ef6c881"
      }
    },
    "rootVolume": {
      "size": 50,
      "volumetype": "SAS"
    },
    "runtime": {
      "name": "docker"
    },
    "login": {
      "sshKey": "KeyPair-001"
    },
    "storage": {
      "storageSelectors": [ {
        "name": "cceUse",
        "storageType": "evs",
        "matchLabels": {
          "size": "100",
          "volumeType": "SAS",

```

```

        "count" : "1"
      }
    },
    "storageGroups" : [ {
      "name" : "vgpaas",
      "selectorNames" : [ "cceUse" ],
      "cceManaged" : true,
      "virtualSpaces" : [ {
        "name" : "runtime",
        "size" : "90%"
      }, {
        "name" : "kubernetes",
        "size" : "10%"
      } ]
    } ]
  } ]
},
"count" : 1
}
}

```

- Create a pay-per-use node with the flavor of 2 vCPUs and 4 GiB of memory. The node runs EulerOS 2.5 for Docker containers. The sizes of the system disk and data disk of the node are 50 GB and 100 GB, respectively. Both the disks are of the high I/O type.

POST /api/v3/projects/{project_id}/clusters/{cluster_id}/nodes

```

{
  "kind" : "Node",
  "apiVersion" : "v3",
  "metadata" : {
    "name" : "test-83790"
  },
  "spec" : {
    "flavor" : "c7.large.2",
    "az" : "*****",
    "os" : "EulerOS 2.5",
    "dataVolumes" : [ {
      "size" : 100,
      "volumetype" : "SAS"
    } ],
    "billingMode" : 0,
    "extendParam" : {
      "maxPods" : 110
    },
    "nodeNicSpec" : {
      "primaryNic" : {
        "subnetId" : "ca964acf-8468-4735-8229-97940ef6c881"
      }
    },
    "rootVolume" : {
      "size" : 50,
      "volumetype" : "SAS"
    },
    "runtime" : {
      "name" : "docker"
    },
    "login" : {
      "sshKey" : "KeyPair-001"
    },
    "storage" : {
      "storageSelectors" : [ {
        "name" : "cceUse",
        "storageType" : "evs",
        "matchLabels" : {
          "size" : "100",
          "volumeType" : "SAS",
          "count" : "1"
        }
      } ]
    }
  }
}

```



```

"storageGroups" : [ {
  "name" : "vgpaas",
  "selectorNames" : [ "cceUse" ],
  "cceManaged" : true,
  "virtualSpaces" : [ {
    "name" : "runtime",
    "size" : "90%"
  }, {
    "name" : "kubernetes",
    "size" : "10%"
  } ]
} ]
},
"count" : 1
}
}

```

- Create a pay-per-usenode with the flavor of 2 vCPUs and 4 GiB of memory. The node runs Huawei Cloud EulerOS 2.0 and uses Docker. The sizes of the system disk and data disk of the node are 50 GB and 100 GB, respectively. The data disk uses the shared disk space mode.

POST /api/v3/projects/{project_id}/clusters/{cluster_id}/nodes

```

{
  "kind" : "Node",
  "apiVersion" : "v3",
  "metadata" : {
    "name" : "test-66909"
  },
  "spec" : {
    "flavor" : "c7.large.2",
    "az" : "*****",
    "os" : "Huawei Cloud EulerOS 2.0",
    "dataVolumes" : [ {
      "size" : 100,
      "volumetype" : "SAS"
    } ],
    "billingMode" : 0,
    "extendParam" : {
      "maxPods" : 110
    },
    "nodeNicSpec" : {
      "primaryNic" : {
        "subnetId" : "ca964acf-8468-4735-8229-97940ef6c881"
      }
    },
    "rootVolume" : {
      "size" : 50,
      "volumetype" : "SAS"
    },
    "runtime" : {
      "name" : "docker"
    },
    "login" : {
      "sshKey" : "KeyPair-001"
    },
    "storage" : {
      "storageSelectors" : [ {
        "name" : "cceUse",
        "storageType" : "evs",
        "matchLabels" : {
          "size" : "100",
          "volumeType" : "SAS",
          "count" : "1"
        }
      } ],
      "storageGroups" : [ {
        "name" : "vgpaas",
        "selectorNames" : [ "cceUse" ],

```

```

    "cceManaged" : true,
    "virtualSpaces" : [ {
      "name" : "share",
      "size" : "100%"
    } ]
  } ]
},
"count" : 1
}
}

```

Example Responses

Status code: 201

The job for creating a node in a specified cluster is successfully delivered.

```

{
  "kind" : "Node",
  "apiVersion" : "v3",
  "metadata" : {
    "name" : "test-83790",
    "uid" : "5ecfddfe-87db-11ec-b5e5-0255ac101514",
    "annotations" : {
      "jobid" : "5ec1518c-87db-11ec-b5e5-0255ac101514",
      "resourceJobId" : "5ed0d692-87db-11ec-b5e5-0255ac101514"
    }
  },
  "spec" : {
    "flavor" : "c7.large.2",
    "az" : "*****",
    "os" : "EulerOS 2.5",
    "login" : {
      "sshKey" : "KeyPair-001"
    },
    "rootVolume" : {
      "volumetype" : "SAS",
      "size" : 50
    },
    "dataVolumes" : [ {
      "volumetype" : "SAS",
      "size" : 100
    } ],
    "storage" : {
      "storageSelectors" : [ {
        "name" : "cceUse",
        "storageType" : "evs",
        "matchLabels" : {
          "count" : "1",
          "size" : "100",
          "volumeType" : "SAS"
        }
      } ],
      "storageGroups" : [ {
        "name" : "vgpaas",
        "cceManaged" : true,
        "selectorNames" : [ "cceUse" ],
        "virtualSpaces" : [ {
          "name" : "runtime",
          "size" : "90%"
        }, {
          "name" : "kubernetes",
          "size" : "10%"
        } ]
      } ]
    },
    "publicIP" : {
      "eip" : {
        "bandwidth" : { }
      }
    }
  }
}

```

```
}
},
"nodeNicSpec" : {
  "primaryNic" : {
    "subnetId" : "ca964acf-8468-4735-8229-97940ef6c881"
  }
},
"count" : 1,
"billingMode" : 0,
"runtime" : {
  "name" : "docker"
},
"extendParam" : {
  "chargingMode" : 0,
  "ecs.performancetype" : "computingv3",
  "init-node-password" : "*****",
  "maxPods" : 110,
  "publicKey" : ""
}
},
"status" : {
  "jobID" : "5ec1518c-87db-11ec-b5e5-0255ac101514"
}
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

- Create a yearly/monthly node with the flavor of 2 vCPUs and 4 GiB of memory. The node runs EulerOS 2.5 for Docker containers. The sizes of the system disk and data disk of the node are 50 GB and 100 GB, respectively. Both the disks are of the high I/O type.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;
import com.huaweicloud.sdk.cce.v3.model.Runtime;

import java.util.List;
import java.util.ArrayList;

public class CreateNodeSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);
```

```
CceClient client = CceClient.newBuilder()
    .withCredential(auth)
    .withRegion(CceRegion.valueOf("<YOUR REGION>"))
    .build();
CreateNodeRequest request = new CreateNodeRequest();
request.withClusterId("{cluster_id}");
NodeCreateRequest body = new NodeCreateRequest();
NodeExtendParam extendParamSpec = new NodeExtendParam();
extendParamSpec.withMaxPods(110)
    .withPeriodType("month")
    .withPeriodNum(1)
    .withIsAutoRenew("false")
    .withIsAutoPay("false");
Runtime runtimeSpec = new Runtime();
runtimeSpec.withName(Runtime.NameEnum.fromValue("docker"));
NicSpec primaryNicNodeNicSpec = new NicSpec();
primaryNicNodeNicSpec.withSubnetId("ca964acf-8468-4735-8229-97940ef6c881");
nodeNicSpecSpec = new NodeNicSpec();
nodeNicSpecSpec.withPrimaryNic(primaryNicNodeNicSpec);
List<VirtualSpace> listStorageGroupsVirtualSpaces = new ArrayList<>();
listStorageGroupsVirtualSpaces.add(
    new VirtualSpace()
        .withName("runtime")
        .withSize("90%")
);
listStorageGroupsVirtualSpaces.add(
    new VirtualSpace()
        .withName("kubernetes")
        .withSize("10%")
);
List<String> listStorageGroupsSelectorNames = new ArrayList<>();
listStorageGroupsSelectorNames.add("cceUse");
List<StorageGroups> listStorageStorageGroups = new ArrayList<>();
listStorageStorageGroups.add(
    new StorageGroups()
        .withName("vgpaas")
        .withCceManaged(true)
        .withSelectorNames(listStorageGroupsSelectorNames)
        .withVirtualSpaces(listStorageGroupsVirtualSpaces)
);
StorageSelectorsMatchLabels matchLabelsStorageSelectors = new
StorageSelectorsMatchLabels();
matchLabelsStorageSelectors.withSize("100")
    .withVolumeType("SAS")
    .withCount("1");
List<StorageSelectors> listStorageStorageSelectors = new ArrayList<>();
listStorageStorageSelectors.add(
    new StorageSelectors()
        .withName("cceUse")
        .withStorageType("evs")
        .withMatchLabels(matchLabelsStorageSelectors)
);
Storage storageSpec = new Storage();
storageSpec.withStorageSelectors(listStorageStorageSelectors)
    .withStorageGroups(listStorageStorageGroups);
List<Volume> listSpecDataVolumes = new ArrayList<>();
listSpecDataVolumes.add(
    new Volume()
        .withSize(100)
        .withVolumetype("SAS")
);
Volume rootVolumeSpec = new Volume();
rootVolumeSpec.withSize(50)
    .withVolumetype("SAS");
Login loginSpec = new Login();
loginSpec.withSshKey("KeyPair-001");
NodeSpec specbody = new NodeSpec();
specbody.withFlavor("c7.large.2")
```

```
.withAz("*****")
.withOs("EulerOS 2.5")
.withLogin(loginSpec)
.withRootVolume(rootVolumeSpec)
.withDataVolumes(listSpecDataVolumes)
.withStorage(storageSpec)
.withNodeNicSpec(nodeNicSpecSpec)
.withCount(1)
.withBillingMode(1)
.withRuntime(runtimeSpec)
.withExtendParam(extendParamSpec);
NodeMetadata metadatabody = new NodeMetadata();
metadatabody.setName("test-67235");
body.withSpec(specbody);
body.withMetadata(metadatabody);
body.withApiVersion("v3");
body.withKind("Node");
request.withBody(body);
try {
    CreateNodeResponse response = client.createNode(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

- Create a pay-per-use node with the flavor of 2 vCPUs and 4 GiB of memory. The node runs EulerOS 2.5 for Docker containers. The sizes of the system disk and data disk of the node are 50 GB and 100 GB, respectively. Both the disks are of the high I/O type.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;
import com.huaweicloud.sdk.cce.v3.model.Runtime;

import java.util.List;
import java.util.ArrayList;

public class CreateNodeSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
```

```
.withAk(ak)
.withSk(sk);

CceClient client = CceClient.newBuilder()
    .withCredential(auth)
    .withRegion(CceRegion.valueOf("<YOUR REGION>"))
    .build();
CreateNodeRequest request = new CreateNodeRequest();
request.withClusterId("{cluster_id}");
NodeCreateRequest body = new NodeCreateRequest();
NodeExtendParam extendParamSpec = new NodeExtendParam();
extendParamSpec.withMaxPods(110);
Runtime runtimeSpec = new Runtime();
runtimeSpec.withName(Runtime.NameEnum.fromValue("docker"));
NicSpec primaryNicNodeNicSpec = new NicSpec();
primaryNicNodeNicSpec.withSubnetId("ca964acf-8468-4735-8229-97940ef6c881");
NodeNicSpec nodeNicSpecSpec = new NodeNicSpec();
nodeNicSpecSpec.withPrimaryNic(primaryNicNodeNicSpec);
List<VirtualSpace> listStorageGroupsVirtualSpaces = new ArrayList<>();
listStorageGroupsVirtualSpaces.add(
    new VirtualSpace()
        .withName("runtime")
        .withSize("90%")
);
listStorageGroupsVirtualSpaces.add(
    new VirtualSpace()
        .withName("kubernetes")
        .withSize("10%")
);
List<String> listStorageGroupsSelectorNames = new ArrayList<>();
listStorageGroupsSelectorNames.add("cceUse");
List<StorageGroups> listStorageStorageGroups = new ArrayList<>();
listStorageStorageGroups.add(
    new StorageGroups()
        .withName("vgpaas")
        .withCceManaged(true)
        .withSelectorNames(listStorageGroupsSelectorNames)
        .withVirtualSpaces(listStorageGroupsVirtualSpaces)
);
StorageSelectorsMatchLabels matchLabelsStorageSelectors = new
StorageSelectorsMatchLabels();
matchLabelsStorageSelectors.withSize("100")
    .withVolumeType("SAS")
    .withCount("1");
List<StorageSelectors> listStorageStorageSelectors = new ArrayList<>();
listStorageStorageSelectors.add(
    new StorageSelectors()
        .withName("cceUse")
        .withStorageType("evs")
        .withMatchLabels(matchLabelsStorageSelectors)
);
Storage storageSpec = new Storage();
storageSpec.withStorageSelectors(listStorageStorageSelectors)
    .withStorageGroups(listStorageStorageGroups);
List<Volume> listSpecDataVolumes = new ArrayList<>();
listSpecDataVolumes.add(
    new Volume()
        .withSize(100)
        .withVolumetype("SAS")
);
Volume rootVolumeSpec = new Volume();
rootVolumeSpec.withSize(50)
    .withVolumetype("SAS");
Login loginSpec = new Login();
loginSpec.withSshKey("KeyPair-001");
NodeSpec specbody = new NodeSpec();
specbody.withFlavor("c7.large.2")
    .withAz("*****")
    .withOs("EulerOS 2.5")
```

```
.withLogin(loginSpec)
.withRootVolume(rootVolumeSpec)
.withDataVolumes(listSpecDataVolumes)
.withStorage(storageSpec)
.withNodeNicSpec(nodeNicSpecSpec)
.withCount(1)
.withBillingMode(0)
.withRuntime(runtimeSpec)
.withExtendParam(extendParamSpec);
NodeMetadata metadatabody = new NodeMetadata();
metadatabody.withName("test-83790");
body.withSpec(specbody);
body.withMetadata(metadatabody);
body.withApiVersion("v3");
body.withKind("Node");
request.withBody(body);
try {
    CreateNodeResponse response = client.createNode(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

- Create a pay-per-usenode with the flavor of 2 vCPUs and 4 GiB of memory. The node runs Huawei Cloud EulerOS 2.0 and uses Docker. The sizes of the system disk and data disk of the node are 50 GB and 100 GB, respectively. The data disk uses the shared disk space mode.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;
import com.huaweicloud.sdk.cce.v3.model.Runtime;

import java.util.List;
import java.util.ArrayList;

public class CreateNodeSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);
```

```
CceClient client = CceClient.newBuilder()
    .withCredential(auth)
    .withRegion(CceRegion.valueOf("<YOUR REGION>"))
    .build();
CreateNodeRequest request = new CreateNodeRequest();
request.withClusterId("{cluster_id}");
NodeCreateRequest body = new NodeCreateRequest();
NodeExtendParam extendParamSpec = new NodeExtendParam();
extendParamSpec.withMaxPods(110);
Runtime runtimeSpec = new Runtime();
runtimeSpec.withName(Runtime.NameEnum.fromValue("docker"));
NicSpec primaryNicNodeNicSpec = new NicSpec();
primaryNicNodeNicSpec.withSubnetId("ca964acf-8468-4735-8229-97940ef6c881");
NodeNicSpec nodeNicSpecSpec = new NodeNicSpec();
nodeNicSpecSpec.withPrimaryNic(primaryNicNodeNicSpec);
List<VirtualSpace> listStorageGroupsVirtualSpaces = new ArrayList<>();
listStorageGroupsVirtualSpaces.add(
    new VirtualSpace()
        .withName("share")
        .withSize("100%")
);
List<String> listStorageGroupsSelectorNames = new ArrayList<>();
listStorageGroupsSelectorNames.add("cceUse");
List<StorageGroups> listStorageStorageGroups = new ArrayList<>();
listStorageStorageGroups.add(
    new StorageGroups()
        .withName("vgpaas")
        .withCceManaged(true)
        .withSelectorNames(listStorageGroupsSelectorNames)
        .withVirtualSpaces(listStorageGroupsVirtualSpaces)
);
StorageSelectorsMatchLabels matchLabelsStorageSelectors = new
StorageSelectorsMatchLabels();
matchLabelsStorageSelectors.withSize("100")
    .withVolumeType("SAS")
    .withCount("1");
List<StorageSelectors> listStorageStorageSelectors = new ArrayList<>();
listStorageStorageSelectors.add(
    new StorageSelectors()
        .withName("cceUse")
        .withStorageType("evs")
        .withMatchLabels(matchLabelsStorageSelectors)
);
Storage storageSpec = new Storage();
storageSpec.withStorageSelectors(listStorageStorageSelectors)
    .withStorageGroups(listStorageStorageGroups);
List<Volume> listSpecDataVolumes = new ArrayList<>();
listSpecDataVolumes.add(
    new Volume()
        .withSize(100)
        .withVolumetype("SAS")
);
Volume rootVolumeSpec = new Volume();
rootVolumeSpec.withSize(50)
    .withVolumetype("SAS");
Login loginSpec = new Login();
loginSpec.withSshKey("KeyPair-001");
NodeSpec specbody = new NodeSpec();
specbody.withFlavor("c7.large.2")
    .withAz("*****")
    .withOs("Huawei Cloud EulerOS 2.0")
    .withLogin(loginSpec)
    .withRootVolume(rootVolumeSpec)
    .withDataVolumes(listSpecDataVolumes)
    .withStorage(storageSpec)
    .withNodeNicSpec(nodeNicSpecSpec)
    .withCount(1)
    .withBillingMode(0)
```



```
        .withRuntime(runtimeSpec)
        .withExtendParam(extendParamSpec);
NodeMetadata metadatabody = new NodeMetadata();
metadatabody.setName("test-66909");
body.withSpec(specbody);
body.withMetadata(metadatabody);
body.withApiVersion("v3");
body.withKind("Node");
request.withBody(body);
try {
    CreateNodeResponse response = client.createNode(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

- Create a yearly/monthly node with the flavor of 2 vCPUs and 4 GiB of memory. The node runs EulerOS 2.5 for Docker containers. The sizes of the system disk and data disk of the node are 50 GB and 100 GB, respectively. Both the disks are of the high I/O type.

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *
```

```
if __name__ == "__main__":
```

```
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
```

```
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"
```

```
    credentials = BasicCredentials(ak, sk, projectId) \
```

```
    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()
```

```
try:
```

```
    request = CreateNodeRequest()
    request.cluster_id = "{cluster_id}"
    extendParamSpec = NodeExtendParam(
        max_pods=110,
        period_type="month",
        period_num=1,
        is_auto_renew="false",
        is_auto_pay="false"
    )
```

```
    runtimeSpec = Runtime(
```

```
    name="docker"
  )
  primaryNicNodeNicSpec = NicSpec(
    subnet_id="ca964acf-8468-4735-8229-97940ef6c881"
  )
  nodeNicSpecSpec = NodeNicSpec(
    primary_nic=primaryNicNodeNicSpec
  )
  listVirtualSpacesStorageGroups = [
    VirtualSpace(
      name="runtime",
      size="90%"
    ),
    VirtualSpace(
      name="kubernetes",
      size="10%"
    )
  ]
  listSelectorNamesStorageGroups = [
    "cceUse"
  ]
  listStorageGroupsStorage = [
    StorageGroups(
      name="vgpaas",
      cce_managed=True,
      selector_names=listSelectorNamesStorageGroups,
      virtual_spaces=listVirtualSpacesStorageGroups
    )
  ]
  matchLabelsStorageSelectors = StorageSelectorsMatchLabels(
    size="100",
    volume_type="SAS",
    count="1"
  )
  listStorageSelectorsStorage = [
    StorageSelectors(
      name="cceUse",
      storage_type="evs",
      match_labels=matchLabelsStorageSelectors
    )
  ]
  storageSpec = Storage(
    storage_selectors=listStorageSelectorsStorage,
    storage_groups=listStorageGroupsStorage
  )
  listDataVolumesSpec = [
    Volume(
      size=100,
      volumetype="SAS"
    )
  ]
  rootVolumeSpec = Volume(
    size=50,
    volumetype="SAS"
  )
  loginSpec = Login(
    ssh_key="KeyPair-001"
  )
  specbody = NodeSpec(
    flavor="c7.large.2",
    az="*****",
    os="EulerOS 2.5",
    login=loginSpec,
    root_volume=rootVolumeSpec,
    data_volumes=listDataVolumesSpec,
    storage=storageSpec,
    node_nic_spec=nodeNicSpecSpec,
    count=1,
    billing_mode=1,
```

```
        runtime=runtimeSpec,
        extend_param=extendParamSpec
    )
    metadatabody = NodeMetadata(
        name="test-67235"
    )
    request.body = NodeCreateRequest(
        spec=specbody,
        metadata=metadatabody,
        api_version="v3",
        kind="Node"
    )
    response = client.create_node(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

- Create a pay-per-use node with the flavor of 2 vCPUs and 4 GiB of memory. The node runs EulerOS 2.5 for Docker containers. The sizes of the system disk and data disk of the node are 50 GB and 100 GB, respectively. Both the disks are of the high I/O type.

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *
```

```
if __name__ == "__main__":
```

```
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
```

```
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"
```

```
    credentials = BasicCredentials(ak, sk, projectId) \
```

```
    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()
```

```
try:
```

```
    request = CreateNodeRequest()
    request.cluster_id = "{cluster_id}"
    extendParamSpec = NodeExtendParam(
        max_pods=110
    )
    runtimeSpec = Runtime(
        name="docker"
    )
    primaryNicNodeNicSpec = NicSpec(
        subnet_id="ca964acf-8468-4735-8229-97940ef6c881"
    )
    nodeNicSpecSpec = NodeNicSpec(
        primary_nic=primaryNicNodeNicSpec
    )
    listVirtualSpacesStorageGroups = [
        VirtualSpace(
            name="runtime",
            size="90%"
        ),
    ],
```

```
VirtualSpace(  
    name="kubernetes",  
    size="10%"  
)  
]  
listSelectorNamesStorageGroups = [  
    "cceUse"  
]  
listStorageGroupsStorage = [  
    StorageGroups(  
        name="vgpaas",  
        cce_managed=True,  
        selector_names=listSelectorNamesStorageGroups,  
        virtual_spaces=listVirtualSpacesStorageGroups  
    )  
]  
matchLabelsStorageSelectors = StorageSelectorsMatchLabels(  
    size="100",  
    volume_type="SAS",  
    count="1"  
)  
listStorageSelectorsStorage = [  
    StorageSelectors(  
        name="cceUse",  
        storage_type="evs",  
        match_labels=matchLabelsStorageSelectors  
    )  
]  
storageSpec = Storage(  
    storage_selectors=listStorageSelectorsStorage,  
    storage_groups=listStorageGroupsStorage  
)  
listDataVolumesSpec = [  
    Volume(  
        size=100,  
        volumetype="SAS"  
    )  
]  
rootVolumeSpec = Volume(  
    size=50,  
    volumetype="SAS"  
)  
loginSpec = Login(  
    ssh_key="KeyPair-001"  
)  
specbody = NodeSpec(  
    flavor="c7.large.2",  
    az="*****",  
    os="EulerOS 2.5",  
    login=loginSpec,  
    root_volume=rootVolumeSpec,  
    data_volumes=listDataVolumesSpec,  
    storage=storageSpec,  
    node_nic_spec=nodeNicSpecSpec,  
    count=1,  
    billing_mode=0,  
    runtime=runtimeSpec,  
    extend_param=extendParamSpec  
)  
metadatabody = NodeMetadata(  
    name="test-83790"  
)  
request.body = NodeCreateRequest(  
    spec=specbody,  
    metadata=metadatabody,  
    api_version="v3",  
    kind="Node"  
)  
response = client.create_node(request)
```

```
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

- Create a pay-per-usenode with the flavor of 2 vCPUs and 4 GiB of memory. The node runs Huawei Cloud EulerOS 2.0 and uses Docker. The sizes of the system disk and data disk of the node are 50 GB and 100 GB, respectively. The data disk uses the shared disk space mode.

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateNodeRequest()
        request.cluster_id = "{cluster_id}"
        extendParamSpec = NodeExtendParam(
            max_pods=110
        )
        runtimeSpec = Runtime(
            name="docker"
        )
        primaryNicNodeNicSpec = NicSpec(
            subnet_id="ca964acf-8468-4735-8229-97940ef6c881"
        )
        nodeNicSpecSpec = NodeNicSpec(
            primary_nic=primaryNicNodeNicSpec
        )
        listVirtualSpacesStorageGroups = [
            VirtualSpace(
                name="share",
                size="100%"
            )
        ]
        listSelectorNamesStorageGroups = [
            "cceUse"
        ]
        listStorageGroupsStorage = [
            StorageGroups(
                name="vgpaas",
                cce_managed=True,
                selector_names=listSelectorNamesStorageGroups,
                virtual_spaces=listVirtualSpacesStorageGroups
            )
        ]
        matchLabelsStorageSelectors = StorageSelectorsMatchLabels(
```

```
        size="100",
        volume_type="SAS",
        count="1"
    )
    listStorageSelectorsStorage = [
        StorageSelectors(
            name="cceUse",
            storage_type="evs",
            match_labels=matchLabelsStorageSelectors
        )
    ]
    storageSpec = Storage(
        storage_selectors=listStorageSelectorsStorage,
        storage_groups=listStorageGroupsStorage
    )
    listDataVolumesSpec = [
        Volume(
            size=100,
            volumetype="SAS"
        )
    ]
    rootVolumeSpec = Volume(
        size=50,
        volumetype="SAS"
    )
    loginSpec = Login(
        ssh_key="KeyPair-001"
    )
    specbody = NodeSpec(
        flavor="c7.large.2",
        az="*****",
        os="Huawei Cloud EulerOS 2.0",
        login=loginSpec,
        root_volume=rootVolumeSpec,
        data_volumes=listDataVolumesSpec,
        storage=storageSpec,
        node_nic_spec=nodeNicSpecSpec,
        count=1,
        billing_mode=0,
        runtime=runtimeSpec,
        extend_param=extendParamSpec
    )
    metadatabody = NodeMetadata(
        name="test-66909"
    )
    request.body = NodeCreateRequest(
        spec=specbody,
        metadata=metadatabody,
        api_version="v3",
        kind="Node"
    )
    response = client.create_node(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

- Create a yearly/monthly node with the flavor of 2 vCPUs and 4 GiB of memory. The node runs EulerOS 2.5 for Docker containers. The sizes of the system disk and data disk of the node are 50 GB and 100 GB, respectively. Both the disks are of the high I/O type.

```
package main
```

```
import (
```

```
"fmt"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateNodeRequest{}
    request.ClusterId = "{cluster_id}"
    maxPodsExtendParam := int32(110)
    periodTypeExtendParam := "month"
    periodNumExtendParam := int32(1)
    isAutoRenewExtendParam := "false"
    isAutoPayExtendParam := "false"
    extendParamSpec := &model.NodeExtendParam{
        MaxPods: &maxPodsExtendParam,
        PeriodType: &periodTypeExtendParam,
        PeriodNum: &periodNumExtendParam,
        IsAutoRenew: &isAutoRenewExtendParam,
        IsAutoPay: &isAutoPayExtendParam,
    }
    nameRuntime := model.GetRuntimeNameEnum().DOCKER
    runtimeSpec := &model.Runtime{
        Name: &nameRuntime,
    }
    subnetIdPrimaryNic := "ca964acf-8468-4735-8229-97940ef6c881"
    primaryNicNodeNicSpec := &model.NicSpec{
        SubnetId: &subnetIdPrimaryNic,
    }
    nodeNicSpecSpec := &model.NodeNicSpec{
        PrimaryNic: primaryNicNodeNicSpec,
    }
    var listVirtualSpacesStorageGroups = []model.VirtualSpace{
        {
            Name: "runtime",
            Size: "90%",
        },
        {
            Name: "kubernetes",
            Size: "10%",
        },
    }
    var listSelectorNamesStorageGroups = []string{
        "cceUse",
    }
    cceManagedStorageGroups := true
}
```

```
var listStorageGroupsStorage = []model.StorageGroups{
    {
        Name: "vgpaas",
        CceManaged: &cceManagedStorageGroups,
        SelectorNames: listSelectorNamesStorageGroups,
        VirtualSpaces: listVirtualSpacesStorageGroups,
    },
}
sizeMatchLabels:= "100"
volumeTypeMatchLabels:= "SAS"
countMatchLabels:= "1"
matchLabelsStorageSelectors := &model.StorageSelectorsMatchLabels{
    Size: &sizeMatchLabels,
    VolumeType: &volumeTypeMatchLabels,
    Count: &countMatchLabels,
}
var listStorageSelectorsStorage = []model.StorageSelectors{
    {
        Name: "cceUse",
        StorageType: "evs",
        MatchLabels: matchLabelsStorageSelectors,
    },
}
storageSpec := &model.Storage{
    StorageSelectors: listStorageSelectorsStorage,
    StorageGroups: listStorageGroupsStorage,
}
var listDataVolumesSpec = []model.Volume{
    {
        Size: int32(100),
        Volumetype: "SAS",
    },
}
rootVolumeSpec := &model.Volume{
    Size: int32(50),
    Volumetype: "SAS",
}
sshKeyLogin:= "KeyPair-001"
loginSpec := &model.Login{
    SshKey: &sshKeyLogin,
}
osSpec:= "EulerOS 2.5"
countSpec:= int32(1)
billingModeSpec:= int32(1)
specbody := &model.NodeSpec{
    Flavor: "c7.large.2",
    Az: "*****",
    Os: &osSpec,
    Login: loginSpec,
    RootVolume: rootVolumeSpec,
    DataVolumes: listDataVolumesSpec,
    Storage: storageSpec,
    NodeNicSpec: nodeNicSpecSpec,
    Count: &countSpec,
    BillingMode: &billingModeSpec,
    Runtime: runtimeSpec,
    ExtendParam: extendParamSpec,
}
nameMetadata:= "test-67235"
metadatabody := &model.NodeMetadata{
    Name: &nameMetadata,
}
request.Body = &model.NodeCreateRequest{
    Spec: specbody,
    Metadata: metadatabody,
    ApiVersion: "v3",
    Kind: "Node",
}
response, err := client.CreateNode(request)
```



```
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

- Create a pay-per-use node with the flavor of 2 vCPUs and 4 GiB of memory. The node runs EulerOS 2.5 for Docker containers. The sizes of the system disk and data disk of the node are 50 GB and 100 GB, respectively. Both the disks are of the high I/O type.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateNodeRequest{}
    request.ClusterId = "{cluster_id}"
    maxPodsExtendParam := int32(110)
    extendParamSpec := &model.NodeExtendParam{
        MaxPods: &maxPodsExtendParam,
    }
    nameRuntime := model.GetRuntimeNameEnum().DOCKER
    runtimeSpec := &model.Runtime{
        Name: &nameRuntime,
    }
    subnetIdPrimaryNic := "ca964acf-8468-4735-8229-97940ef6c881"
    primaryNicNodeNicSpec := &model.NicSpec{
        SubnetId: &subnetIdPrimaryNic,
    }
    nodeNicSpecSpec := &model.NodeNicSpec{
        PrimaryNic: primaryNicNodeNicSpec,
    }
    var listVirtualSpacesStorageGroups = []model.VirtualSpace{
        {
            Name: "runtime",
            Size: "90%",
        },
        {
            Name: "kubernetes",
            Size: "10%",
        }
    }
}
```

```
    },
  }
  var listSelectorNamesStorageGroups = []string{
    "cceUse",
  }
  cceManagedStorageGroups:= true
  var listStorageGroupsStorage = []model.StorageGroups{
    {
      Name: "vgpaas",
      CceManaged: &cceManagedStorageGroups,
      SelectorNames: listSelectorNamesStorageGroups,
      VirtualSpaces: listVirtualSpacesStorageGroups,
    },
  }
  sizeMatchLabels:= "100"
  volumeTypeMatchLabels:= "SAS"
  countMatchLabels:= "1"
  matchLabelsStorageSelectors := &model.StorageSelectorsMatchLabels{
    Size: &sizeMatchLabels,
    VolumeType: &volumeTypeMatchLabels,
    Count: &countMatchLabels,
  }
  var listStorageSelectorsStorage = []model.StorageSelectors{
    {
      Name: "cceUse",
      StorageType: "evs",
      MatchLabels: matchLabelsStorageSelectors,
    },
  }
  storageSpec := &model.Storage{
    StorageSelectors: listStorageSelectorsStorage,
    StorageGroups: listStorageGroupsStorage,
  }
  var listDataVolumesSpec = []model.Volume{
    {
      Size: int32(100),
      Volumetype: "SAS",
    },
  }
  rootVolumeSpec := &model.Volume{
    Size: int32(50),
    Volumetype: "SAS",
  }
  sshKeyLogin:= "KeyPair-001"
  loginSpec := &model.Login{
    SshKey: &sshKeyLogin,
  }
  osSpec:= "EulerOS 2.5"
  countSpec:= int32(1)
  billingModeSpec:= int32(0)
  specbody := &model.NodeSpec{
    Flavor: "c7.large.2",
    Az: "*****",
    Os: &osSpec,
    Login: loginSpec,
    RootVolume: rootVolumeSpec,
    DataVolumes: listDataVolumesSpec,
    Storage: storageSpec,
    NodeNicSpec: nodeNicSpecSpec,
    Count: &countSpec,
    BillingMode: &billingModeSpec,
    Runtime: runtimeSpec,
    ExtendParam: extendParamSpec,
  }
  nameMetadata:= "test-83790"
  metadatabody := &model.NodeMetadata{
    Name: &nameMetadata,
  }
  request.Body = &model.NodeCreateRequest{
```

```

    Spec: specbody,
    Metadata: metadatabody,
    ApiVersion: "v3",
    Kind: "Node",
  }
  response, err := client.CreateNode(request)
  if err == nil {
    fmt.Printf("%+v\n", response)
  } else {
    fmt.Println(err)
  }
}

```

- Create a pay-per-usenode with the flavor of 2 vCPUs and 4 GiB of memory. The node runs Huawei Cloud EulerOS 2.0 and uses Docker. The sizes of the system disk and data disk of the node are 50 GB and 100 GB, respectively. The data disk uses the shared disk space mode.

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateNodeRequest{}
    request.ClusterId = "{cluster_id}"
    maxPodsExtendParam := int32(110)
    extendParamSpec := &model.NodeExtendParam{
        MaxPods: &maxPodsExtendParam,
    }
    nameRuntime := model.GetRuntimeNameEnum().DOCKER
    runtimeSpec := &model.Runtime{
        Name: &nameRuntime,
    }
    subnetIdPrimaryNic := "ca964acf-8468-4735-8229-97940ef6c881"
    primaryNicNodeNicSpec := &model.NicSpec{
        SubnetId: &subnetIdPrimaryNic,
    }
    nodeNicSpecSpec := &model.NodeNicSpec{
        PrimaryNic: primaryNicNodeNicSpec,
    }
    var listVirtualSpacesStorageGroups = []model.VirtualSpace{
        {

```

```

        Name: "share",
        Size: "100%",
    },
}
var listSelectorNamesStorageGroups = []string{
    "cceUse",
}
cceManagedStorageGroups:= true
var listStorageGroupsStorage = []model.StorageGroups{
    {
        Name: "vgpaas",
        CceManaged: &cceManagedStorageGroups,
        SelectorNames: listSelectorNamesStorageGroups,
        VirtualSpaces: listVirtualSpacesStorageGroups,
    },
}
sizeMatchLabels:= "100"
volumeTypeMatchLabels:= "SAS"
countMatchLabels:= "1"
matchLabelsStorageSelectors := &model.StorageSelectorsMatchLabels{
    Size: &sizeMatchLabels,
    VolumeType: &volumeTypeMatchLabels,
    Count: &countMatchLabels,
}
var listStorageSelectorsStorage = []model.StorageSelectors{
    {
        Name: "cceUse",
        StorageType: "evs",
        MatchLabels: matchLabelsStorageSelectors,
    },
}
storageSpec := &model.Storage{
    StorageSelectors: listStorageSelectorsStorage,
    StorageGroups: listStorageGroupsStorage,
}
var listDataVolumesSpec = []model.Volume{
    {
        Size: int32(100),
        Volumetype: "SAS",
    },
}
rootVolumeSpec := &model.Volume{
    Size: int32(50),
    Volumetype: "SAS",
}
sshKeyLogin:= "KeyPair-001"
loginSpec := &model.Login{
    SshKey: &sshKeyLogin,
}
osSpec:= "Huawei Cloud EulerOS 2.0"
countSpec:= int32(1)
billingModeSpec:= int32(0)
specbody := &model.NodeSpec{
    Flavor: "c7.large.2",
    Az: "*****",
    Os: &osSpec,
    Login: loginSpec,
    RootVolume: rootVolumeSpec,
    DataVolumes: listDataVolumesSpec,
    Storage: storageSpec,
    NodeNicSpec: nodeNicSpecSpec,
    Count: &countSpec,
    BillingMode: &billingModeSpec,
    Runtime: runtimeSpec,
    ExtendParam: extendParamSpec,
}
nameMetadata:= "test-66909"
metadatabody := &model.NodeMetadata{
    Name: &nameMetadata,
}

```

```
}
request.Body = &model.NodeCreateRequest{
    Spec: specbody,
    Metadata: metadatabody,
    ApiVersion: "v3",
    Kind: "Node",
}
response, err := client.CreateNode(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
201	The job for creating a node in a specified cluster is successfully delivered.

Error Codes

See [Error Codes](#).

4.3.2 Reading a Specified Node

Function

This API is used to obtain details about a specified node via the node ID.

NOTE

The URL for cluster management is in the format of `https://Endpoint/uri`. In the URL, **uri** indicates the resource path, that is, the path for API access.

Calling Method

For details, see [Calling APIs](#).

URI

GET /api/v3/projects/{project_id}/clusters/{cluster_id}/nodes/{node_id}

Table 4-279 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
node_id	Yes	String	Node ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-280 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-281 Response body parameters

Parameter	Type	Description
kind	String	API type. The value is fixed at Node and cannot be changed.
apiVersion	String	API version. The value is fixed at v3 and cannot be changed.

Parameter	Type	Description
metadata	NodeMetadata object	Node metadata, which is a collection of attributes.
spec	NodeSpec object	Detailed description of the node. CCE creates or updates objects by defining or updating spec .
status	NodeStatus object	Node status, which is dynamically recorded. A user-defined value will not function when a node is being created or modified.

Table 4-282 NodeMetadata

Parameter	Type	Description
name	String	<p>Node name</p> <p>NOTE Enter 1 to 56 characters starting with a letter and not ending with a hyphen (-). Only lowercase letters, digits, and hyphens (-) are allowed.</p> <p>If name is left empty or is not specified, the node name is generated based on the default rule, which is <i>Cluster name-Random characters</i>. Only the first 36 characters are kept for a long cluster name.</p> <p>If the number of nodes (count) is greater than 1, some random characters are added to the end of the entered node name according to the default rule, which is <i>User-defined name-Random characters</i>. Only the first 50 characters in the custom name are kept, followed by some random characters.</p> <p>Names of nodes running in clusters v1.28.1, v1.27.3, v1.25.6, v1.23.11, v1.21.12 or later support periods (.), but this configuration is not recommended.</p>
uid	String	Node ID, which is unique and automatically generated after the resource is created. A user-defined ID will not take effect.
labels	Map<String,String>	<p>CCE node label (not the native Kubernetes label).</p> <p>Labels are used to select objects that meet certain criteria. A label is a key-value pair.</p> <p>Example:</p> <pre>"labels": { "key": "value" }</pre>

Parameter	Type	Description
annotations	Map<String,String>	<p>CCE node annotations, in key-value pairs (not the native Kubernetes annotations).</p> <p>Example:</p> <pre>"annotations": { "key1" : "value1", "key2" : "value2" }</pre> <p>NOTE</p> <ul style="list-style-type: none"> • annotations are not used to identify or select objects. The metadata in annotations may be small or large, structured or unstructured, and may include characters that are not allowed in labels. • This parameter is used only for query and cannot be input through a request. Entered data of this parameter is invalid.
creationTimestamp	String	Time when the object was created. The value is automatically generated after the object is created. A user-defined value will not take effect.
updateTimestamp	String	Time when the object was updated. The value is automatically generated after the object is created. A user-defined value will not take effect.

Table 4-283 NodeSpec

Parameter	Type	Description
flavor	String	Node specifications. For details about the node specifications supported by CCE, see Node Flavor Description .
az	String	<p>AZ where the node to be created is located. You need to specify the AZ name. If it is set to random, the node will be created in a random AZ.</p> <p>For details about AZs supported by CCE, see Regions and Endpoints.</p>

Parameter	Type	Description
os	String	<p>Node OS. For details about the supported OSs, see Node OS.</p> <p>NOTE</p> <ul style="list-style-type: none"> The system automatically selects the supported OS based on the cluster version. If the current cluster version does not support the OS, an error will be reported. If <code>alpha.cce/NodeImageID</code> in <code>extendParam</code> is specified during node creation, you do not need to configure this parameter. This parameter is mandatory when creating a node pool. If you are using shared disk space when creating a node, this parameter is mandatory. Specifically, the disk initialization configuration management parameter <code>storage</code> is used and the <code>name</code> field of <code>virtualSpaces</code> in <code>StorageGroups</code> is set to <code>share</code>.
login	Login object	Node login mode. Either the key pair or password must be used for login.
rootVolume	Volume object	Information about disks on the node
dataVolumes	Array of Volume objects	<p>Data disk parameters of the node. Currently, you can add the second data disk for your node on the CCE console.</p> <p>This data disk is used by the container runtime and kubelet. Do not uninstall this disk. Otherwise, the node will become unavailable.</p> <p>For DeC nodes, the parameter description is the same as that for <code>rootVolume</code>.</p>

Parameter	Type	Description
storage	Storage object	<p>Disk initialization management parameter. This parameter is complex to configure. For details, see Attaching Disks to a Node.</p> <p>If this parameter retains its default, disks are managed based on the DockerLVMConfigOverride (discarded) parameter in extendParam. This parameter is supported by clusters of version 1.15.11 and later.</p> <p>NOTE</p> <ul style="list-style-type: none"> • If a node specification involves both local disks and EVS disks, do not retain the default value of this parameter to prevent unexpected disk partitions. • If you want to change the value range of a data disk to 20 to 32768, do not use the default value. • If you want to use the shared disk space (with the runtime and Kubernetes partitions cancelled), do not retain the default value of this parameter. For details about the shared disk space, see Data Disk Space Allocation.
publicIP	NodePublicIP object	<p>EIP of a node.</p> <p>NOTE This parameter is not supported when you add a node to a node pool.</p>
nodeNicSpec	NodeNicSpec object	NIC of the node
count	Integer	Number of nodes to be created in a batch. The value must be a positive integer greater than or equal to 1 and less than or equal to the defined limit. This parameter can be left blank when it is used for a node pool.
billingMode	Integer	<p>Node billing mode.</p> <ul style="list-style-type: none"> • 0: pay-per-use • 1: yearly/monthly • 2: (discarded) yearly/monthly billing with auto payment enabled

Parameter	Type	Description
taints	Array of Taint objects	<p>You can add taints to created nodes to configure anti-affinity. A maximum of 20 taints can be added. Each taint contains the following parameters:</p> <ul style="list-style-type: none"> • Key: A key must contain 1 to 63 characters starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain name can be used as the prefix of a key. • Value: A value must start with a letter or digit and can contain a maximum of 63 characters, including letters, digits, hyphens (-), underscores (_), and periods (.). • Effect: Available options are NoSchedule, PreferNoSchedule, and NoExecute. When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query. <p>Example:</p> <pre>"taints": [{ "key": "status", "value": "unavailable", "effect": "NoSchedule" }, { "key": "looks", "value": "bad", "effect": "NoSchedule" }]</pre>

Parameter	Type	Description
k8sTags	Map<String,String>	<p>Defined in key-value pairs. A maximum of 20 key-value pairs are allowed.</p> <ul style="list-style-type: none"> Key: Enter 1 to 63 characters, starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain can be prefixed to a key and contain a maximum of 253 characters. Example DNS subdomain: example.com/my-key Value: The value can be left blank or contain 1 to 63 characters that start with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed in the character string. When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query. <p>Example:</p> <pre>"k8sTags": { "key": "value" }</pre>
ecsGroupId	String	<p>ECS group ID. If this parameter is configured, nodes will be created in the specified ECS group.</p> <p>NOTE This configuration does not take effect when a node pool is created. To ensure that all nodes in a node pool are in the same ECS group, configure nodeManagement in the node pool.</p>
dedicatedHostId	String	<p>ID of the DeH to which the node is scheduled.</p> <p>NOTE This parameter is not supported when you add a node during node pool creation.</p>

Parameter	Type	Description
userTags	Array of UserTag objects	<p>Cloud server tag. The key of a tag must be unique. The maximum number of custom tags supported by CCE depends on the region and cannot exceed 8.</p> <p>When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query.</p> <p>NOTE Use only letters, Unicode characters, digits, special characters(-, _). Max characters: 36</p>
runtime	Runtime object	<p>Container runtime:</p> <ul style="list-style-type: none"> • Clusters of v1.25 or earlier: docker. • Clusters of v1.25 or later: Container runtime varies with the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker. For nodes running other OSs, the default container runtime is containerd.

Parameter	Type	Description
initializedConditions	Array of strings	<p>Custom initialization flag, which is left blank by default.</p> <p>Before CCE nodes are initialized, node.cloudprovider.kubernetes.io/uninitialized is added to the nodes to prevent pods from being scheduled to these nodes. When creating a node, you can configure the initializedConditions parameter to control the taint removal time. By default, the timeout period is not configured.</p> <p>Example:</p> <ol style="list-style-type: none"> 1. Create a node and add parameter "initializedConditions": ["CCEInitial", "CustomedInitial"]. 2. After custom initialization is complete, CCE will call a Kubernetes API (for example, PATCH /v1/nodes/{node_ip}/status) to update the node conditions by adding two labels of types CCEInitial and CustomedInitial and setting the status to True, as shown in the following: <pre data-bbox="815 1055 1430 1205"> status: conditions: - type: CCEInitial status: 'True' - type: CustomedInitial status: 'True' </pre> <ol style="list-style-type: none"> 1. CCE polls status.Conditions of nodes to check whether there are conditions of types CCEInitial and CustomedInitial. If such conditions exist and the status is True, the node initialization is complete and the initialization taint is removed. 2. initializedConditions allows you to configure a timeout period during node creation. For example, "initializedConditions": ["CCEInitial:15m", "CustomedInitial:15m"] indicates that the timeout period is 15 minutes. After the timeout period is reached, the initialization conditions will be automatically ignored and the initialization taint will be removed when CCE polls the node. <ul style="list-style-type: none"> ● Use only letters and digits. Max. characters: 20. ● Max. flags: 2. ● The unit of the timeout period is minute (m).

Parameter	Type	Description
extendParam	NodeExtendParam object	Extended parameters for creating a node.
hostnameConfig	HostnameConfig object	Kubernetes node name configuration parameter, which is supported by clusters of v1.23.6-r0 to v1.25 or clusters of v1.25.2-r0 or later versions.

Table 4-284 Login

Parameter	Type	Description
sshKey	String	Name of the key pair used for login.
userPassword	UserPassword object	Password used for node login.

Table 4-285 UserPassword

Parameter	Type	Description
username	String	Login account. The default value is root .
password	String	<p>If a username and a password are used to create a node, this field is shielded in the response body.</p> <p>A password must meet the following complexity requirements:</p> <ul style="list-style-type: none"> • Contains 8 to 26 characters. • Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (!@\$%^-_=+[{ } ; , / ? ~ # *) • Cannot contain the username or the username spelled backwards. <p>The password field must be salted during node creation. For details, see Adding a Salt in the password Field When Creating a Node.</p>

Table 4-286 Volume

Parameter	Type	Description
size	Integer	<p>Disk size, in GB.</p> <ul style="list-style-type: none"> • System disk: 40 to 1024 • Value range for the first data disk: 20 to 32768 (When storage is not specified, this value range becomes 100 to 32768.) • Value range for other data disks: 10 to 32768. (When storage is not specified, this value range becomes 100 to 32768.)
volumetype	String	<p>Disk type. For details about possible values, see the description of the root_volume parameter in the API used to create an ECS.</p> <ul style="list-style-type: none"> • SAS: high I/O SAS disk • SSD: ultra-high I/O SSD disk • SATA: common I/O SATA disk SATA disks have been removed from EVS. You can find them attached only on existing nodes.
extendParam	Map<String, Object>	<p>Extended disk parameters, defined in extendparam in the API used to create an ECS. For details, see Creating an ECS.</p>
cluster_id	String	<p>ID of the storage pool used by the ECS system disk. This field is used only for DeC clusters, which functions as dssPoolID, that is, the ID of the DSS storage pool.</p> <p>To obtain the value, see the ID field in the API used to obtain details of a DSS storage pool.</p>
cluster_type	String	<p>Storage class of the cloud server system disk. The value is always dss. This field is used only for DeC clusters.</p>
hw:passthrough	Boolean	<ul style="list-style-type: none"> • Pay attention to this field if your ECS is SDI-compliant. If the value of this field is true, an SCSI disk will be created. • If the node pool type is ElasticBMS, this field must be set to true. • If a node specification involves local disks and EVS disks at the same time, set the disk initialization parameters. For details, see Attaching Disks to a Node.
metadata	VolumeMetadata object	<p>EVS disk encryption information. This field is mandatory only when you need to encrypt the system disk or data disks of the node to be created.</p>

Table 4-287 VolumeMetadata

Parameter	Type	Description
__system__encrypted	String	Whether the EVS disk is encrypted. The value 0 indicates that the EVS disk is not encrypted, and the value 1 indicates that the EVS disk is encrypted. If this parameter is not specified, EVS disks will not be encrypted by default.
__system__cmkid	String	CMK ID, which indicates encryption in metadata . This field is used with __system__encrypted .

Table 4-288 Storage

Parameter	Type	Description
storageSelectors	Array of StorageSelectors objects	Disk selection. Matched disks are managed according to matchLabels and storageType .
storageGroups	Array of StorageGroups objects	A storage group consists of multiple storage devices. It is used to divide storage space.

Table 4-289 StorageSelectors

Parameter	Type	Description
name	String	Selector name, used as the index of selectorNames in storageGroup . Therefore, the name of each selector must be unique.
storageType	String	Specifies the storage type. Currently, only evs (EVS volumes) and local (local volumes) are supported. The local storage does not support disk selection. All local disks will form a VG. Therefore, only one storageSelector of the local type is allowed.
matchLabels	matchLabels object	Matching field of an EVS volume. The size , volumeType , metadataEncrypted , metadataCmkid and count fields are supported.

Table 4-290 matchLabels

Parameter	Type	Description
size	String	Matched disk size. If this parameter is left unspecified, the disk size is not limited. Example: 100
volumeType	String	EVS disk type. SSD, GPSSD, SAS, ESSD, and SATA are supported. If this parameter is left blank, there is no restriction on the disk type.
metadataEncrypted	String	Disk encryption identifier. If this parameter is set to 0 , it specifies that the disk is not encrypted. If this parameter is set to 1 , it specifies that the disk is encrypted. If this parameter is left blank, there is no restriction on the disk encryption identifier.
metadataCmkid	String	CMK ID of an encrypted disk. The value is a string of 36 bytes. If this parameter is left blank, there is no restriction on the disk key ID.
count	String	Number of disks to be selected. If this parameter is left blank, all disks of this type are selected.

Table 4-291 StorageGroups

Parameter	Type	Description
name	String	Name of a virtual storage group, which must be unique. NOTE <ul style="list-style-type: none"> If cceManaged is set to true, the name must be set to vgpaas. If the data disk is used as a temporary storage volume, the name must be vg-everest-localvolume-ephemeral. If the data disk is used as a persistent storage volume, the name must be vg-everest-localvolume-persistent.
cceManaged	Boolean	Storage space for Kubernetes and runtime components. Only one group can be set to true . If this parameter is left blank, the default value false is used.
selectorNames	Array of strings	This parameter corresponds to name in storageSelectors . A group can match multiple selectors, but a selector can match only one group.

Parameter	Type	Description
virtualSpaces	Array of VirtualSpace objects	Detailed management of space configuration in a group.

Table 4-292 VirtualSpace

Parameter	Type	Description
name	String	Name of a virtualSpace. Options: share , kubernetes , runtime , and user <ul style="list-style-type: none"> • share: shared space configuration (with the runtime and Kubernetes partitions cancelled). lvmConfig needs to be configured. • Kubernetes: Kubernetes space configuration. lvmConfig needs to be configured. • runtime: runtime space configuration. runtimeConfig needs to be configured. • user: user space configuration. lvmConfig needs to be configured.
size	String	Size of a virtualSpace. The value must be an integer in percentage. Example: 90%. NOTE The sum of the percentages of all virtualSpaces in a group cannot exceed 100%.
lvmConfig	LVMConfig object	LVM configuration managements, applicable to the <i>share</i> *, kubernetes , and user spaces. Note that one virtual space supports only one config.
runtimeConfig	RuntimeConfig object	runtime configurations, applicable to the runtime space. Note that one virtual space supports only one config.

Table 4-293 LVMConfig

Parameter	Type	Description
lvType	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.

Parameter	Type	Description
path	String	Path to which the disk is attached. This parameter takes effect only in user configuration. The value is an absolute path. Digits, letters, periods (.), hyphens (-), and underscores (_) are allowed.

Table 4-294 RuntimeConfig

Parameter	Type	Description
lvType	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.

Table 4-295 NodePublicIP

Parameter	Type	Description
ids	Array of strings	IDs of existing EIPs. The quantity cannot be greater than the number of nodes to be created. NOTE If ids has been set, you do not need to set count and eip .
count	Integer	Number of EIPs to be dynamically created. NOTE count and eip must be set at the same time.
eip	NodeEIPSpec object	EIP configuration.

Table 4-296 NodeEIPSpec

Parameter	Type	Description
iptype	String	EIP type, specified in publicip.type in the API for assigning an EIP. For details, see Assigning an EIP .
bandwidth	NodeBandwidth object	Bandwidth parameters of the EIP

Table 4-297 NodeBandwidth

Parameter	Type	Description
chargemode	String	<p>Bandwidth billing mode.</p> <ul style="list-style-type: none"> • If this field is not specified, the billing is based on bandwidth. • If the field is null, the billing is based on bandwidth. • If the field value is traffic, the billing is based on traffic. • If the value is out of the preceding options, the cloud server will fail to be created. <p>NOTE</p> <ul style="list-style-type: none"> • Billed by bandwidth: The billing will be based on the data transmission rate (in Mbps) of public networks. This billing mode is recommended if your bandwidth usage is higher than 10%. • Billed by traffic: The billing is based on the total amount of data (in GB) transmitted over the public network. This mode is available only when you are creating a pay-per-use node. This billing mode is recommended if your bandwidth usage is lower than 10%.
size	Integer	<p>Bandwidth size, specified in bandwidth.size in the API for assigning an EIP.</p> <p>For details, see Assigning an EIP.</p>
sharetype	String	<p>Bandwidth sharing type. Value options: PER (exclusive bandwidth)</p>

Table 4-298 NodeNicSpec

Parameter	Type	Description
primaryNic	NicSpec object	Description of the primary NIC.
extNics	Array of NicSpec objects	<p>Extension NIC</p> <p>NOTE This parameter is not supported when you add a node to a node pool.</p>

Table 4-299 NicSpec

Parameter	Type	Description
subnetId	String	Network ID of the subnet to which a NIC belongs. If subnetId is not specified when a primary NIC is creating, the cluster subnet will be used. If subnetList is also configured for a node pool, the subnetList field is used for adding subnets to the node pool. When creating an extension NIC, you must specify subnetId .
fixedIps	Array of strings	The IP address of the primary ENI is specified using fixedIps . The number of IP addresses cannot be greater than the number of created nodes. Either fixedIps or ipBlock can be specified. fixedIps cannot be specified for extension ENIs.
ipBlock	String	CIDR format of the primary NIC IP range. The IP address of the created node falls in this range. fixedIps and ipBlock cannot be specified at the same time.

Table 4-300 Taint

Parameter	Type	Description
key	String	Key.
value	String	Value.
effect	String	Effect.

Table 4-301 UserTag

Parameter	Type	Description
key	String	Key of the cloud server label. The value cannot start with CCE- or __type_baremetal .
value	String	Value of the cloud server label.

Table 4-302 Runtime

Parameter	Type	Description
name	String	<p>Container runtime. Default value:</p> <ul style="list-style-type: none"> • Clusters earlier than v1.25: The default value is docker. • Clusters of v1.25 or later: The default container runtime varies depending on the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker; for nodes running other OSs, the default container runtime is containerd.

Table 4-303 NodeExtendParam

Parameter	Type	Description
ecs:performancetype	String	ECS flavor types. This field is returned in the response.
orderId	String	Order ID. This parameter is returned in the response when the node is billed on a yearly/monthly basis with auto payment enabled (only in creation scenarios).
productId	String	Product ID. This field is returned in the response when the node is yearly/monthly-billed with auto payment enabled.
maxPods	Integer	<p>Maximum number of pods that can be created on a node, including the default system pods. Value range: 16 to 256.</p> <p>This limit prevents the node from being overloaded of pods.</p> <p>The number of pods that can be created on a node is determined by multiple parameters. For details, see Maximum Number of Pods That Can Be Created on a Node.</p>
periodType	String	<ul style="list-style-type: none"> • month: The unit is month. • year: The unit is year. <p>NOTE</p> <p>Request parameter, which is valid and mandatory only when billingMode is set to 1 (yearly/monthly billing) or 2 (yearly/monthly billing with auto renewal enabled) (discarded).</p> <p>Response parameter, which is returned only when a yearly/monthly node is created.</p>

Parameter	Type	Description
periodNum	Integer	<p>Subscription duration. The value can be:</p> <ul style="list-style-type: none"> • If periodType is month, the value ranges from 1 to 9. • If periodType is year, the value is 1. <p>NOTE Request parameter, which is valid and mandatory when billingMode is set to 1 or 2 (discarded). Response parameter, which is returned only when a yearly/monthly node is created.</p>
isAutoRenew	String	<p>Whether auto renewal is enabled.</p> <ul style="list-style-type: none"> • true: Auto renewal is enabled. • false: Auto renewal is not enabled. <p>NOTE This field is valid when billingMode is set to 1 or 2 (discarded). If not specified, auto renewal is not enabled.</p>
isAutoPay	String	<p>Whether to deduct fees automatically.</p> <ul style="list-style-type: none"> • true: Enable automatic fee deduction. • false: Do not enable automatic fee deduction. <p>NOTE This field is valid when billingMode is set to 1 or 2 (discarded). If not specified when billingMode is set to 1, auto fee deduction is not enabled. (Discarded: If not specified when billingMode is set to 2, auto fee deduction is enabled.)</p>

Parameter	Type	Description
DockerLVMConfigOverride	String	<p>Docker data disk configuration item. (This parameter has been discarded. Use the storage field instead.) Example default configuration: "DockerLVMConfigOverride":"dockerThinpool=vgpaas/90%VG;kubernetesLV=vgpaas/10%VG;diskType=evs;lvType=linear"</p> <p>By default, if no VD disk is available, an error occurs because the data disk fails to be found. Set diskType based on the actual drive letter type.</p> <p>The following fields are included:</p> <ul style="list-style-type: none"> ● userLV (optional): size of the user space, for example, vgpaas/20%VG. ● userPath (optional): mount path of the user space, for example, /home/wqt-test. ● diskType: disk type. Currently, only evs, hdd, and ssd are supported. ● lvType: type of a logic volume. The value can be linear or striped. ● dockerThinpool: Docker space size, for example, vgpaas/60%VG. ● kubernetesLV: kubelet space size, for example, vgpaas/20%VG.
dockerBaseSize	Integer	<p>Available disk space of a single container on a node, in GB. This parameter has been discarded. Use containerBaseSize.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In Device Mapper mode, the default value is 10. In OverlayFS mode, the available space of a single container is not limited by default, and the dockerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters of the new versions.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that dockerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device.</p>

Parameter	Type	Description
containerBaseSize	Integer	<p>Available disk space of a single container on a node, in GB.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In OverlayFS mode, the available space of a single container is not limited by default. In Device Mapper mode, the default value is 10, and the containerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters v1.23.14-r0, v1.25.9-r0, v1.27.6-r0, v1.28.4-r0 or later.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that containerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device. In new versions, the Device Mapper mode is used only by BMSs in the same resource pool and is being discarded.</p>
publicKey	String	Public key of a node.
alpha.cce/preInstall	String	<p>Pre-installation script.</p> <p>NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)</p>
alpha.cce/postInstall	String	<p>Post-installation script.</p> <p>NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)</p>
alpha.cce/NodeImageID	String	This parameter is required when a custom image is used to create a BMS node.

Parameter	Type	Description
nicMultiqueue	String	<ul style="list-style-type: none"> Number of ENI queues. Example setting: <code>"[{"queue":4}]"</code> The following fields are included: <ul style="list-style-type: none"> queue: number of ENI queues. This field can be configured only for BMS nodes in a CCE Turbo cluster. Supported proportions are {"1":128, "2":92, "4":92, "8":32, "16":16, "28":9}. That is, if there is one queue, a maximum of 128 ENIs can be bound. If there are two queues, a maximum of 92 ENIs can be bound for two queues. A larger number of ENI queues indicates higher performance but fewer ENIs can be bound. The queue settings cannot be changed after the node pool is created.
nicThreshold	String	<ul style="list-style-type: none"> ENI pre-binding thresholds. Example setting: <code>"0.3:0.6"</code> <ul style="list-style-type: none"> Low threshold (L): determines the minimum number of pre-bound ENIs (Min). Formula: $Min = Total\ number\ of\ ENIs\ of\ the\ node \times L$ High threshold (H): determines the maximum number of pre-bound ENIs (Max). Formula: $Max = Total\ number\ of\ ENIs\ of\ the\ node \times H$ Number of ENIs bound to a BMS node (B) and number of ENIs being used by pods (U): $U + Min < B < U + Max$ If the number of pre-bound ENIs on a BMS node is smaller than the minimum allowed, the system will bind more ENIs to make the numbers equal. If the number of pre-bound ENIs on a BMS node is larger than the maximum allowed, the system periodically unbinds ENIs (about every 2 minutes) to make the numbers equal. Both the thresholds are one-decimal-place values ranging from 0.0 to 1.0. The low threshold must be smaller than or equal to the high one. This field can be configured only for BMS nodes in a CCE Turbo cluster. Pre-binding ENIs can speed up workload creation but occupies IP addresses.

Parameter	Type	Description
chargingMode	Integer	Billing mode of a node. This parameter has been deprecated. Use the billingMode parameter in NodeSpec .
agency_name	String	Name of an agency An agency is created by a tenant administrator on Identity and Access Management (IAM) to provide temporary credentials for CCE nodes to access cloud servers. This parameter is returned only when it is transferred during node creation.
kubeReservedMem	Integer	Reserved node memory, which is reserved for Kubernetes components.
systemReservedMem	Integer	Reserved node memory, which is reserved for system components.
init-node-password	String	Node password. If this parameter is used as a response parameter, asterisks (*) will be displayed.

Table 4-304 HostnameConfig

Parameter	Type	Description
type	String	Configuration type of the Kubernetes node name. The default value is privatelp . <ul style="list-style-type: none"> • privatelp: The Kubernetes node is named after its IP address. • cceNodeName: The Kubernetes node is named after the CCE node. <p>NOTE</p> <ul style="list-style-type: none"> • For a node which is configured using cceNodeName, the name is the same as the Kubernetes node name and the ECS name. The node name cannot be changed. If the ECS name is changed on the ECS console, the node name will retain unchanged after ECS synchronization. • For a node which is configured using cceNodeName, to avoid a conflict between Kubernetes nodes, the system automatically adds a suffix to each node name. The suffix is in the format of A hyphen (-) Five random characters. The value of the random characters is a lowercase letter or a digit ranging from 0 to 9.

Table 4-305 NodeStatus

Parameter	Type	Description
phase	String	Node status, which is the status of a node during its lifecycle (such as installation and uninstallation) or the status of a node in a Kubernetes cluster. Options: <ul style="list-style-type: none"> ● Build: The node is being created. ● Installing: The node is being managed. ● Upgrading: The node is being upgraded. ● Active: The node is running properly. ● Abnormal: The node is abnormal. ● Deleting: The node is being deleted. ● Error: The node is faulty.
lastProbeTime	String	Last time when the node status was checked. If the cluster is in the abnormal, frozen, or intermediate state (for example, creating), the node status check may be affected. The node status that takes more than five minutes to check has no reference value.
jobID	String	ID of a creation or deletion job
serverId	String	ID of the underlying ECS or BMS node
privateIP	String	IP address in the private network segment of the primary NIC on the node
privateIPv6IP	String	IPv6 address in the private network segment of the primary NIC on the node
publicIP	String	Node EIP. If the ECS data is not synchronized in real time, you can click Sync Node Data on the console to manually update the data.
deleteStatus	DeleteStatus object	Resource status during resource deletion.

Table 4-306 DeleteStatus

Parameter	Type	Description
previous_total	Integer	Total number of existing cluster resource records when the cluster is deleted.
current_total	Integer	Latest number of resource records, which is generated based on the current cluster resource records.

Parameter	Type	Description
updated	Integer	Total number of resource records updated when the cluster is deleted.
added	Integer	Total number of resource records updated when the cluster is deleted.
deleted	Integer	Total number of resource records deleted when the cluster is deleted.

Example Requests

None

Example Responses

Status code: 200

Information about the specified node is successfully obtained.

```
{
  "kind": "Node",
  "apiVersion": "v3",
  "metadata": {
    "name": "myhost",
    "uid": "4d1ecb2c-229a-11e8-9c75-0255ac100ceb",
    "creationTimestamp": "2018-08-02 08:12:40.124294439 +0000 UTC",
    "updateTimestamp": "2018-08-02 08:18:20.221871842 +0000 UTC",
    "annotations": {
      "kubernetes.io/node-pool.id": "az1.dc1#s1.medium#EulerOS 2.2"
    }
  },
  "spec": {
    "flavor": "s1.medium",
    "az": "*****",
    "os": "EulerOS 2.2",
    "login": {
      "sshKey": "KeyPair-001"
    },
    "rootVolume": {
      "volumetype": "SAS",
      "size": 40
    },
    "dataVolumes": [ {
      "volumetype": "SAS",
      "size": 100
    } ],
    "publicIP": {
      "eip": {
        "bandwidth": { }
      }
    },
    "billingMode": 0
  },
  "status": {
    "phase": "Active",
    "serverId": "456789abc-9368-46f3-8f29-d1a95622a568",
    "publicIP": "10.34.56.78",
    "privateIP": "192.168.1.23"
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ShowNodeSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowNodeRequest request = new ShowNodeRequest();
        request.withClusterId("{cluster_id}");
        request.withNodeId("{node_id}");
        try {
            ShowNodeResponse response = client.showNode(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *
```

```

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowNodeRequest()
        request.cluster_id = "{cluster_id}"
        request.node_id = "{node_id}"
        response = client.show_node(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)

```

Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowNodeRequest{}
    request.ClusterId = "{cluster_id}"
    request.NodeId = "{node_id}"
    response, err := client.ShowNode(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    }
}

```



```

    } else {
      fmt.Println(err)
    }
  }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Information about the specified node is successfully obtained.

Error Codes

See [Error Codes](#).

4.3.3 Listing All Nodes in a Cluster

Function

This API is used to obtain details about all nodes in a specified cluster with the cluster ID.

NOTE

The URL for cluster management is in the format of `https://Endpoint/uri`. In the URL, **uri** indicates the resource path, that is, the path for API access.

Calling Method

For details, see [Calling APIs](#).

URI

GET `/api/v3/projects/{project_id}/clusters/{cluster_id}/nodes`

Table 4-307 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Parameter	Mandatory	Type	Description
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-308 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-309 Response body parameters

Parameter	Type	Description
kind	String	API type. The value is fixed at List .
apiVersion	String	API version. The value is fixed at v3 .
items	Array of Node objects	List of details for all nodes in the current cluster. You can filter nodes by items.metadata.name .

Table 4-310 Node

Parameter	Type	Description
kind	String	API type. The value is fixed at Node and cannot be changed.

Parameter	Type	Description
apiVersion	String	API version. The value is fixed at v3 and cannot be changed.
metadata	NodeMetadata object	Node metadata, which is a collection of attributes.
spec	NodeSpec object	Detailed description of the node. CCE creates or updates objects by defining or updating spec .
status	NodeStatus object	Node status, which is dynamically recorded. A user-defined value will not function when a node is being created or modified.

Table 4-311 NodeMetadata

Parameter	Type	Description
name	String	<p>Node name</p> <p>NOTE Enter 1 to 56 characters starting with a letter and not ending with a hyphen (-). Only lowercase letters, digits, and hyphens (-) are allowed.</p> <p>If name is left empty or is not specified, the node name is generated based on the default rule, which is <i>Cluster name-Random characters</i>. Only the first 36 characters are kept for a long cluster name.</p> <p>If the number of nodes (count) is greater than 1, some random characters are added to the end of the entered node name according to the default rule, which is <i>User-defined name-Random characters</i>. Only the first 50 characters in the custom name are kept, followed by some random characters.</p> <p>Names of nodes running in clusters v1.28.1, v1.27.3, v1.25.6, v1.23.11, v1.21.12 or later support periods (.), but this configuration is not recommended.</p>
uid	String	Node ID, which is unique and automatically generated after the resource is created. A user-defined ID will not take effect.
labels	Map<String,String>	<p>CCE node label (not the native Kubernetes label).</p> <p>Labels are used to select objects that meet certain criteria. A label is a key-value pair.</p> <p>Example:</p> <pre>"labels": { "key" : "value" }</pre>

Parameter	Type	Description
annotations	Map<String,String>	<p>CCE node annotations, in key-value pairs (not the native Kubernetes annotations).</p> <p>Example:</p> <pre>"annotations": { "key1" : "value1", "key2" : "value2" }</pre> <p>NOTE</p> <ul style="list-style-type: none"> annotations are not used to identify or select objects. The metadata in annotations may be small or large, structured or unstructured, and may include characters that are not allowed in labels. This parameter is used only for query and cannot be input through a request. Entered data of this parameter is invalid.
creationTimestamp	String	Time when the object was created. The value is automatically generated after the object is created. A user-defined value will not take effect.
updateTimestamp	String	Time when the object was updated. The value is automatically generated after the object is created. A user-defined value will not take effect.

Table 4-312 NodeSpec

Parameter	Type	Description
flavor	String	Node specifications. For details about the node specifications supported by CCE, see Node Flavor Description .
az	String	<p>AZ where the node to be created is located. You need to specify the AZ name. If it is set to random, the node will be created in a random AZ.</p> <p>For details about AZs supported by CCE, see Regions and Endpoints.</p>

Parameter	Type	Description
os	String	<p>Node OS. For details about the supported OSs, see Node OS.</p> <p>NOTE</p> <ul style="list-style-type: none"> The system automatically selects the supported OS based on the cluster version. If the current cluster version does not support the OS, an error will be reported. If <code>alpha.cce/NodeImageID</code> in <code>extendParam</code> is specified during node creation, you do not need to configure this parameter. This parameter is mandatory when creating a node pool. If you are using shared disk space when creating a node, this parameter is mandatory. Specifically, the disk initialization configuration management parameter <code>storage</code> is used and the <code>name</code> field of <code>virtualSpaces</code> in <code>StorageGroups</code> is set to <code>share</code>.
login	Login object	Node login mode. Either the key pair or password must be used for login.
rootVolume	Volume object	Information about disks on the node
dataVolumes	Array of Volume objects	<p>Data disk parameters of the node. Currently, you can add the second data disk for your node on the CCE console.</p> <p>This data disk is used by the container runtime and kubelet. Do not uninstall this disk. Otherwise, the node will become unavailable.</p> <p>For DeC nodes, the parameter description is the same as that for <code>rootVolume</code>.</p>

Parameter	Type	Description
storage	Storage object	<p>Disk initialization management parameter. This parameter is complex to configure. For details, see Attaching Disks to a Node.</p> <p>If this parameter retains its default, disks are managed based on the DockerLVMConfigOverride (discarded) parameter in extendParam. This parameter is supported by clusters of version 1.15.11 and later.</p> <p>NOTE</p> <ul style="list-style-type: none"> • If a node specification involves both local disks and EVS disks, do not retain the default value of this parameter to prevent unexpected disk partitions. • If you want to change the value range of a data disk to 20 to 32768, do not use the default value. • If you want to use the shared disk space (with the runtime and Kubernetes partitions cancelled), do not retain the default value of this parameter. For details about the shared disk space, see Data Disk Space Allocation.
publicIP	NodePublicIP object	<p>EIP of a node.</p> <p>NOTE This parameter is not supported when you add a node to a node pool.</p>
nodeNicSpec	NodeNicSpec object	NIC of the node
count	Integer	Number of nodes to be created in a batch. The value must be a positive integer greater than or equal to 1 and less than or equal to the defined limit. This parameter can be left blank when it is used for a node pool.
billingMode	Integer	<p>Node billing mode.</p> <ul style="list-style-type: none"> • 0: pay-per-use • 1: yearly/monthly • 2: (discarded) yearly/monthly billing with auto payment enabled

Parameter	Type	Description
taints	Array of Taint objects	<p>You can add taints to created nodes to configure anti-affinity. A maximum of 20 taints can be added. Each taint contains the following parameters:</p> <ul style="list-style-type: none"> • Key: A key must contain 1 to 63 characters starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain name can be used as the prefix of a key. • Value: A value must start with a letter or digit and can contain a maximum of 63 characters, including letters, digits, hyphens (-), underscores (_), and periods (.). • Effect: Available options are NoSchedule, PreferNoSchedule, and NoExecute. When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query. <p>Example:</p> <pre>"taints": [{ "key": "status", "value": "unavailable", "effect": "NoSchedule" }, { "key": "looks", "value": "bad", "effect": "NoSchedule" }]</pre>

Parameter	Type	Description
k8sTags	Map<String,String>	<p>Defined in key-value pairs. A maximum of 20 key-value pairs are allowed.</p> <ul style="list-style-type: none"> Key: Enter 1 to 63 characters, starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain can be prefixed to a key and contain a maximum of 253 characters. Example DNS subdomain: example.com/my-key Value: The value can be left blank or contain 1 to 63 characters that start with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed in the character string. When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query. <p>Example:</p> <pre>"k8sTags": { "key": "value" }</pre>
ecsGroupId	String	<p>ECS group ID. If this parameter is configured, nodes will be created in the specified ECS group.</p> <p>NOTE This configuration does not take effect when a node pool is created. To ensure that all nodes in a node pool are in the same ECS group, configure nodeManagement in the node pool.</p>
dedicatedHostId	String	<p>ID of the DeH to which the node is scheduled.</p> <p>NOTE This parameter is not supported when you add a node during node pool creation.</p>

Parameter	Type	Description
userTags	Array of UserTag objects	<p>Cloud server tag. The key of a tag must be unique. The maximum number of custom tags supported by CCE depends on the region and cannot exceed 8.</p> <p>When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query.</p> <p>NOTE Use only letters, Unicode characters, digits, special characters(-, _). Max characters: 36</p>
runtime	Runtime object	<p>Container runtime:</p> <ul style="list-style-type: none"> • Clusters of v1.25 or earlier: docker. • Clusters of v1.25 or later: Container runtime varies with the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker. For nodes running other OSs, the default container runtime is containerd.

Parameter	Type	Description
initializedConditions	Array of strings	<p>Custom initialization flag, which is left blank by default.</p> <p>Before CCE nodes are initialized, node.cloudprovider.kubernetes.io/uninitialized is added to the nodes to prevent pods from being scheduled to these nodes. When creating a node, you can configure the initializedConditions parameter to control the taint removal time. By default, the timeout period is not configured.</p> <p>Example:</p> <ol style="list-style-type: none"> 1. Create a node and add parameter "initializedConditions": ["CCEInitial", "CustomedInitial"]. 2. After custom initialization is complete, CCE will call a Kubernetes API (for example, PATCH /v1/nodes/{node_ip}/status) to update the node conditions by adding two labels of types CCEInitial and CustomedInitial and setting the status to True, as shown in the following: <pre data-bbox="818 1055 1430 1205"> status: conditions: - type: CCEInitial status: 'True' - type: CustomedInitial status: 'True' </pre> <ol style="list-style-type: none"> 1. CCE polls status.Conditions of nodes to check whether there are conditions of types CCEInitial and CustomedInitial. If such conditions exist and the status is True, the node initialization is complete and the initialization taint is removed. 2. initializedConditions allows you to configure a timeout period during node creation. For example, "initializedConditions": ["CCEInitial:15m", "CustomedInitial:15m"] indicates that the timeout period is 15 minutes. After the timeout period is reached, the initialization conditions will be automatically ignored and the initialization taint will be removed when CCE polls the node. <ul style="list-style-type: none"> ● Use only letters and digits. Max. characters: 20. ● Max. flags: 2. ● The unit of the timeout period is minute (m).

Parameter	Type	Description
extendParam	NodeExtendParam object	Extended parameters for creating a node.
hostnameConfig	HostnameConfig object	Kubernetes node name configuration parameter, which is supported by clusters of v1.23.6-r0 to v1.25 or clusters of v1.25.2-r0 or later versions.

Table 4-313 Login

Parameter	Type	Description
sshKey	String	Name of the key pair used for login.
userPassword	UserPassword object	Password used for node login.

Table 4-314 UserPassword

Parameter	Type	Description
username	String	Login account. The default value is root .
password	String	<p>If a username and a password are used to create a node, this field is shielded in the response body.</p> <p>A password must meet the following complexity requirements:</p> <ul style="list-style-type: none"> • Contains 8 to 26 characters. • Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (!@\$%^-_=+[]{};,:./?~#*) • Cannot contain the username or the username spelled backwards. <p>The password field must be salted during node creation. For details, see Adding a Salt in the password Field When Creating a Node.</p>

Table 4-315 Volume

Parameter	Type	Description
size	Integer	Disk size, in GB. <ul style="list-style-type: none"> System disk: 40 to 1024 Value range for the first data disk: 20 to 32768 (When storage is not specified, this value range becomes 100 to 32768.) Value range for other data disks: 10 to 32768. (When storage is not specified, this value range becomes 100 to 32768.)
volumetype	String	Disk type. For details about possible values, see the description of the root_volume parameter in the API used to create an ECS. <ul style="list-style-type: none"> SAS: high I/O SAS disk SSD: ultra-high I/O SSD disk SATA: common I/O SATA disk SATA disks have been removed from EVS. You can find them attached only on existing nodes.
extendParam	Map<String, Object>	Extended disk parameters, defined in extendparam in the API used to create an ECS. For details, see Creating an ECS .
cluster_id	String	ID of the storage pool used by the ECS system disk. This field is used only for DeC clusters, which functions as dssPoolID , that is, the ID of the DSS storage pool. To obtain the value, see the ID field in the API used to obtain details of a DSS storage pool .
cluster_type	String	Storage class of the cloud server system disk. The value is always dss . This field is used only for DeC clusters.
hw:passthrough	Boolean	<ul style="list-style-type: none"> Pay attention to this field if your ECS is SDI-compliant. If the value of this field is true, an SCSI disk will be created. If the node pool type is ElasticBMS, this field must be set to true. If a node specification involves local disks and EVS disks at the same time, set the disk initialization parameters. For details, see Attaching Disks to a Node.
metadata	VolumeMetadata object	EVS disk encryption information. This field is mandatory only when you need to encrypt the system disk or data disks of the node to be created.

Table 4-316 VolumeMetadata

Parameter	Type	Description
<code>__system__encrypted</code>	String	Whether the EVS disk is encrypted. The value 0 indicates that the EVS disk is not encrypted, and the value 1 indicates that the EVS disk is encrypted. If this parameter is not specified, EVS disks will not be encrypted by default.
<code>__system__cmkid</code>	String	CMK ID, which indicates encryption in metadata . This field is used with __system__encrypted .

Table 4-317 Storage

Parameter	Type	Description
<code>storageSelectors</code>	Array of StorageSelectors objects	Disk selection. Matched disks are managed according to matchLabels and storageType .
<code>storageGroups</code>	Array of StorageGroups objects	A storage group consists of multiple storage devices. It is used to divide storage space.

Table 4-318 StorageSelectors

Parameter	Type	Description
<code>name</code>	String	Selector name, used as the index of selectorNames in storageGroup . Therefore, the name of each selector must be unique.
<code>storageType</code>	String	Specifies the storage type. Currently, only evs (EVS volumes) and local (local volumes) are supported. The local storage does not support disk selection. All local disks will form a VG. Therefore, only one storageSelector of the local type is allowed.
<code>matchLabels</code>	matchLabels object	Matching field of an EVS volume. The size , volumeType , metadataEncrypted , metadataCmkid and count fields are supported.

Table 4-319 matchLabels

Parameter	Type	Description
size	String	Matched disk size. If this parameter is left unspecified, the disk size is not limited. Example: 100
volumeType	String	EVS disk type. SSD, GPSSD, SAS, ESSD, and SATA are supported. If this parameter is left blank, there is no restriction on the disk type.
metadataEncrypted	String	Disk encryption identifier. If this parameter is set to 0 , it specifies that the disk is not encrypted. If this parameter is set to 1 , it specifies that the disk is encrypted. If this parameter is left blank, there is no restriction on the disk encryption identifier.
metadataCmkid	String	CMK ID of an encrypted disk. The value is a string of 36 bytes. If this parameter is left blank, there is no restriction on the disk key ID.
count	String	Number of disks to be selected. If this parameter is left blank, all disks of this type are selected.

Table 4-320 StorageGroups

Parameter	Type	Description
name	String	Name of a virtual storage group, which must be unique. NOTE <ul style="list-style-type: none"> If cceManaged is set to true, the name must be set to vgpaas. If the data disk is used as a temporary storage volume, the name must be vg-everest-localvolume-ephemeral. If the data disk is used as a persistent storage volume, the name must be vg-everest-localvolume-persistent.
cceManaged	Boolean	Storage space for Kubernetes and runtime components. Only one group can be set to true . If this parameter is left blank, the default value false is used.
selectorNames	Array of strings	This parameter corresponds to name in storageSelectors . A group can match multiple selectors, but a selector can match only one group.

Parameter	Type	Description
virtualSpaces	Array of VirtualSpace objects	Detailed management of space configuration in a group.

Table 4-321 VirtualSpace

Parameter	Type	Description
name	String	Name of a virtualSpace. Options: share , kubernetes , runtime , and user <ul style="list-style-type: none"> • share: shared space configuration (with the runtime and Kubernetes partitions cancelled). lvmConfig needs to be configured. • Kubernetes: Kubernetes space configuration. lvmConfig needs to be configured. • runtime: runtime space configuration. runtimeConfig needs to be configured. • user: user space configuration. lvmConfig needs to be configured.
size	String	Size of a virtualSpace. The value must be an integer in percentage. Example: 90%. NOTE The sum of the percentages of all virtualSpaces in a group cannot exceed 100%.
lvmConfig	LVMConfig object	LVM configuration managements, applicable to the <i>share</i> *, kubernetes , and user spaces. Note that one virtual space supports only one config.
runtimeConfig	RuntimeConfig object	runtime configurations, applicable to the runtime space. Note that one virtual space supports only one config.

Table 4-322 LVMConfig

Parameter	Type	Description
lvType	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.

Parameter	Type	Description
path	String	Path to which the disk is attached. This parameter takes effect only in user configuration. The value is an absolute path. Digits, letters, periods (.), hyphens (-), and underscores (_) are allowed.

Table 4-323 RuntimeConfig

Parameter	Type	Description
lvType	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.

Table 4-324 NodePublicIP

Parameter	Type	Description
ids	Array of strings	IDs of existing EIPs. The quantity cannot be greater than the number of nodes to be created. NOTE If ids has been set, you do not need to set count and eip .
count	Integer	Number of EIPs to be dynamically created. NOTE count and eip must be set at the same time.
eip	NodeEIPSpec object	EIP configuration.

Table 4-325 NodeEIPSpec

Parameter	Type	Description
iptype	String	EIP type, specified in publicip.type in the API for assigning an EIP. For details, see Assigning an EIP .
bandwidth	NodeBandwidth object	Bandwidth parameters of the EIP

Table 4-326 NodeBandwidth

Parameter	Type	Description
chargemode	String	<p>Bandwidth billing mode.</p> <ul style="list-style-type: none"> • If this field is not specified, the billing is based on bandwidth. • If the field is null, the billing is based on bandwidth. • If the field value is traffic, the billing is based on traffic. • If the value is out of the preceding options, the cloud server will fail to be created. <p>NOTE</p> <ul style="list-style-type: none"> • Billed by bandwidth: The billing will be based on the data transmission rate (in Mbps) of public networks. This billing mode is recommended if your bandwidth usage is higher than 10%. • Billed by traffic: The billing is based on the total amount of data (in GB) transmitted over the public network. This mode is available only when you are creating a pay-per-use node. This billing mode is recommended if your bandwidth usage is lower than 10%.
size	Integer	<p>Bandwidth size, specified in bandwidth.size in the API for assigning an EIP.</p> <p>For details, see Assigning an EIP.</p>
sharetype	String	<p>Bandwidth sharing type. Value options: PER (exclusive bandwidth)</p>

Table 4-327 NodeNicSpec

Parameter	Type	Description
primaryNic	NicSpec object	Description of the primary NIC.
extNics	Array of NicSpec objects	<p>Extension NIC</p> <p>NOTE This parameter is not supported when you add a node to a node pool.</p>

Table 4-328 NicSpec

Parameter	Type	Description
subnetId	String	Network ID of the subnet to which a NIC belongs. If subnetId is not specified when a primary NIC is creating, the cluster subnet will be used. If subnetList is also configured for a node pool, the subnetList field is used for adding subnets to the node pool. When creating an extension NIC, you must specify subnetId .
fixedIps	Array of strings	The IP address of the primary ENI is specified using fixedIps . The number of IP addresses cannot be greater than the number of created nodes. Either fixedIps or ipBlock can be specified. fixedIps cannot be specified for extension ENIs.
ipBlock	String	CIDR format of the primary NIC IP range. The IP address of the created node falls in this range. fixedIps and ipBlock cannot be specified at the same time.

Table 4-329 Taint

Parameter	Type	Description
key	String	Key.
value	String	Value.
effect	String	Effect.

Table 4-330 UserTag

Parameter	Type	Description
key	String	Key of the cloud server label. The value cannot start with CCE- or __type_baremetal .
value	String	Value of the cloud server label.

Table 4-331 Runtime

Parameter	Type	Description
name	String	<p>Container runtime. Default value:</p> <ul style="list-style-type: none"> • Clusters earlier than v1.25: The default value is docker. • Clusters of v1.25 or later: The default container runtime varies depending on the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker; for nodes running other OSs, the default container runtime is containerd.

Table 4-332 NodeExtendParam

Parameter	Type	Description
ecs:performancetype	String	ECS flavor types. This field is returned in the response.
orderId	String	Order ID. This parameter is returned in the response when the node is billed on a yearly/monthly basis with auto payment enabled (only in creation scenarios).
productId	String	Product ID. This field is returned in the response when the node is yearly/monthly-billed with auto payment enabled.
maxPods	Integer	<p>Maximum number of pods that can be created on a node, including the default system pods. Value range: 16 to 256.</p> <p>This limit prevents the node from being overloaded of pods.</p> <p>The number of pods that can be created on a node is determined by multiple parameters. For details, see Maximum Number of Pods That Can Be Created on a Node.</p>
periodType	String	<ul style="list-style-type: none"> • month: The unit is month. • year: The unit is year. <p>NOTE</p> <p>Request parameter, which is valid and mandatory only when billingMode is set to 1 (yearly/monthly billing) or 2 (yearly/monthly billing with auto renewal enabled) (discarded).</p> <p>Response parameter, which is returned only when a yearly/monthly node is created.</p>

Parameter	Type	Description
periodNum	Integer	<p>Subscription duration. The value can be:</p> <ul style="list-style-type: none"> • If periodType is month, the value ranges from 1 to 9. • If periodType is year, the value is 1. <p>NOTE Request parameter, which is valid and mandatory when billingMode is set to 1 or 2 (discarded). Response parameter, which is returned only when a yearly/monthly node is created.</p>
isAutoRenew	String	<p>Whether auto renewal is enabled.</p> <ul style="list-style-type: none"> • true: Auto renewal is enabled. • false: Auto renewal is not enabled. <p>NOTE This field is valid when billingMode is set to 1 or 2 (discarded). If not specified, auto renewal is not enabled.</p>
isAutoPay	String	<p>Whether to deduct fees automatically.</p> <ul style="list-style-type: none"> • true: Enable automatic fee deduction. • false: Do not enable automatic fee deduction. <p>NOTE This field is valid when billingMode is set to 1 or 2 (discarded). If not specified when billingMode is set to 1, auto fee deduction is not enabled. (Discarded: If not specified when billingMode is set to 2, auto fee deduction is enabled.)</p>

Parameter	Type	Description
DockerLVMConfigOverride	String	<p>Docker data disk configuration item. (This parameter has been discarded. Use the storage field instead.) Example default configuration: <code>"DockerLVMConfigOverride":"dockerThinpool=vgpaas/90%VG;kubernetesLV=vgpaas/10%VG;diskType=evs;lvType=linear"</code></p> <p>By default, if no VD disk is available, an error occurs because the data disk fails to be found. Set diskType based on the actual drive letter type.</p> <p>The following fields are included:</p> <ul style="list-style-type: none"> • userLV (optional): size of the user space, for example, vgpaas/20%VG. • userPath (optional): mount path of the user space, for example, /home/wqt-test. • diskType: disk type. Currently, only evs, hdd, and ssd are supported. • lvType: type of a logic volume. The value can be linear or striped. • dockerThinpool: Docker space size, for example, vgpaas/60%VG. • kubernetesLV: kubelet space size, for example, vgpaas/20%VG.
dockerBaseSize	Integer	<p>Available disk space of a single container on a node, in GB. This parameter has been discarded. Use containerBaseSize.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In Device Mapper mode, the default value is 10. In OverlayFS mode, the available space of a single container is not limited by default, and the dockerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters of the new versions.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that dockerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device.</p>

Parameter	Type	Description
containerBaseSize	Integer	<p>Available disk space of a single container on a node, in GB.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In OverlayFS mode, the available space of a single container is not limited by default. In Device Mapper mode, the default value is 10, and the containerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters v1.23.14-r0, v1.25.9-r0, v1.27.6-r0, v1.28.4-r0 or later.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that containerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device. In new versions, the Device Mapper mode is used only by BMSs in the same resource pool and is being discarded.</p>
publicKey	String	Public key of a node.
alpha.cce/preInstall	String	<p>Pre-installation script.</p> <p>NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)</p>
alpha.cce/postInstall	String	<p>Post-installation script.</p> <p>NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)</p>
alpha.cce/NodeImageID	String	This parameter is required when a custom image is used to create a BMS node.

Parameter	Type	Description
nicMultiqueue	String	<ul style="list-style-type: none"> Number of ENI queues. Example setting: <code>"[{"queue":4}]"</code> The following fields are included: <ul style="list-style-type: none"> queue: number of ENI queues. This field can be configured only for BMS nodes in a CCE Turbo cluster. Supported proportions are {"1":128, "2":92, "4":92, "8":32, "16":16, "28":9}. That is, if there is one queue, a maximum of 128 ENIs can be bound. If there are two queues, a maximum of 92 ENIs can be bound for two queues. A larger number of ENI queues indicates higher performance but fewer ENIs can be bound. The queue settings cannot be changed after the node pool is created.
nicThreshold	String	<ul style="list-style-type: none"> ENI pre-binding thresholds. Example setting: <code>"0.3:0.6"</code> <ul style="list-style-type: none"> Low threshold (L): determines the minimum number of pre-bound ENIs (Min). Formula: $Min = Total\ number\ of\ ENIs\ of\ the\ node \times L$ High threshold (H): determines the maximum number of pre-bound ENIs (Max). Formula: $Max = Total\ number\ of\ ENIs\ of\ the\ node \times H$ Number of ENIs bound to a BMS node (B) and number of ENIs being used by pods (U): $U + Min < B < U + Max$ If the number of pre-bound ENIs on a BMS node is smaller than the minimum allowed, the system will bind more ENIs to make the numbers equal. If the number of pre-bound ENIs on a BMS node is larger than the maximum allowed, the system periodically unbinds ENIs (about every 2 minutes) to make the numbers equal. Both the thresholds are one-decimal-place values ranging from 0.0 to 1.0. The low threshold must be smaller than or equal to the high one. This field can be configured only for BMS nodes in a CCE Turbo cluster. Pre-binding ENIs can speed up workload creation but occupies IP addresses.

Parameter	Type	Description
chargingMode	Integer	Billing mode of a node. This parameter has been deprecated. Use the billingMode parameter in NodeSpec .
agency_name	String	Name of an agency An agency is created by a tenant administrator on Identity and Access Management (IAM) to provide temporary credentials for CCE nodes to access cloud servers. This parameter is returned only when it is transferred during node creation.
kubeReservedMem	Integer	Reserved node memory, which is reserved for Kubernetes components.
systemReservedMem	Integer	Reserved node memory, which is reserved for system components.
init-node-password	String	Node password. If this parameter is used as a response parameter, asterisks (*) will be displayed.

Table 4-333 HostnameConfig

Parameter	Type	Description
type	String	Configuration type of the Kubernetes node name. The default value is privatelp . <ul style="list-style-type: none"> • privatelp: The Kubernetes node is named after its IP address. • cceNodeName: The Kubernetes node is named after the CCE node. <p>NOTE</p> <ul style="list-style-type: none"> • For a node which is configured using cceNodeName, the name is the same as the Kubernetes node name and the ECS name. The node name cannot be changed. If the ECS name is changed on the ECS console, the node name will retain unchanged after ECS synchronization. • For a node which is configured using cceNodeName, to avoid a conflict between Kubernetes nodes, the system automatically adds a suffix to each node name. The suffix is in the format of A hyphen (-) Five random characters. The value of the random characters is a lowercase letter or a digit ranging from 0 to 9.

Table 4-334 NodeStatus

Parameter	Type	Description
phase	String	Node status, which is the status of a node during its lifecycle (such as installation and uninstallation) or the status of a node in a Kubernetes cluster. Options: <ul style="list-style-type: none"> • Build: The node is being created. • Installing: The node is being managed. • Upgrading: The node is being upgraded. • Active: The node is running properly. • Abnormal: The node is abnormal. • Deleting: The node is being deleted. • Error: The node is faulty.
lastProbeTime	String	Last time when the node status was checked. If the cluster is in the abnormal, frozen, or intermediate state (for example, creating), the node status check may be affected. The node status that takes more than five minutes to check has no reference value.
jobID	String	ID of a creation or deletion job
serverId	String	ID of the underlying ECS or BMS node
privateIP	String	IP address in the private network segment of the primary NIC on the node
privateIPv6IP	String	IPv6 address in the private network segment of the primary NIC on the node
publicIP	String	Node EIP. If the ECS data is not synchronized in real time, you can click Sync Node Data on the console to manually update the data.
deleteStatus	DeleteStatus object	Resource status during resource deletion.

Table 4-335 DeleteStatus

Parameter	Type	Description
previous_total	Integer	Total number of existing cluster resource records when the cluster is deleted.
current_total	Integer	Latest number of resource records, which is generated based on the current cluster resource records.

Parameter	Type	Description
updated	Integer	Total number of resource records updated when the cluster is deleted.
added	Integer	Total number of resource records updated when the cluster is deleted.
deleted	Integer	Total number of resource records deleted when the cluster is deleted.

Example Requests

None

Example Responses

Status code: 200

Information about all the nodes in the specified cluster is successfully obtained.

```
{
  "kind": "List",
  "apiVersion": "v3",
  "items": [ {
    "kind": "Node",
    "apiVersion": "v3",
    "metadata": {
      "name": "myhost",
      "uid": "4d1ecb2c-229a-11e8-9c75-0255ac100ceb",
      "creationTimestamp": "2018-08-02 07:37:24.005071325 +0000 UTC",
      "updateTimestamp": "2018-08-02 07:44:04.965500815 +0000 UTC",
      "annotations": {
        "kubernetes.io/node-pool.id": "az1.dc1#s1.medium#EulerOS 2.2"
      }
    },
    "spec": {
      "flavor": "s1.medium",
      "az": "az1.dc1",
      "os": "EulerOS 2.2",
      "login": {
        "sshKey": "KeyPair-001"
      },
      "rootVolume": {
        "volumetype": "SAS",
        "size": 40
      },
      "dataVolumes": [ {
        "volumetype": "SAS",
        "size": 100
      } ],
      "publicIP": {
        "eip": {
          "bandwidth": { }
        }
      }
    },
    "billingMode": 0
  },
  "status": {
    "phase": "Active",
    "serverId": "456789abc-9368-46f3-8f29-d1a95622a568",
    "publicIP": "10.34.56.78",
  }
}
```

```
"privateIP" : "192.168.1.23"  
  }  
} ]  
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cce.v3.region.CceRegion;  
import com.huaweicloud.sdk.cce.v3.*;  
import com.huaweicloud.sdk.cce.v3.model.*;  
  
public class ListNodesSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        CceClient client = CceClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListNodesRequest request = new ListNodesRequest();  
        request.withClusterId("{cluster_id}");  
        try {  
            ListNodesResponse response = client.listNodes(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

Python

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListNodesRequest()
        request.cluster_id = "{cluster_id}"
        response = client.list_nodes(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListNodesRequest{}
    request.ClusterId = "{cluster_id}"
```

```
response, err := client.ListNodes(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Information about all the nodes in the specified cluster is successfully obtained.

Error Codes

See [Error Codes](#).

4.3.4 Updating a Specified Node

Function

This API is used to update information about a specified node.

NOTE

- Currently, only the **name** field in **metadata** can be updated. This field indicates the node name.
- The URL for cluster management is in the format of **https://Endpoint/uri**. In the URL, **uri** indicates the resource path, that is, the path for API access.

Calling Method

For details, see [Calling APIs](#).

URI

PUT /api/v3/projects/{project_id}/clusters/{cluster_id}/nodes/{node_id}

Table 4-336 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
node_id	Yes	String	Node ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-337 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Table 4-338 Request body parameters

Parameter	Mandatory	Type	Description
metadata	Yes	ClusterNodeInformationMetadata object	Node metadata, which is a collection of attributes.

Table 4-339 ClusterNodeInformationMetadata

Parameter	Mandatory	Type	Description
name	Yes	String	<p>Node name</p> <p>NOTE</p> <p>After the node name is changed, the ECS name (VM name) will be changed accordingly.</p> <p>Enter 1 to 56 characters starting with a letter and not ending with a hyphen (-). Only lowercase letters, digits, and hyphens (-) are allowed.</p> <p>Names of nodes running in clusters v1.28.1, v1.27.3, v1.25.6, v1.23.11, v1.21.12 or later support periods (.), but this configuration is not recommended.</p>

Response Parameters

Status code: 200

Table 4-340 Response body parameters

Parameter	Type	Description
kind	String	API type. The value is fixed at Node and cannot be changed.
apiVersion	String	API version. The value is fixed at v3 and cannot be changed.
metadata	NodeMetadata object	Node metadata, which is a collection of attributes.
spec	NodeSpec object	Detailed description of the node. CCE creates or updates objects by defining or updating spec .
status	NodeStatus object	Node status, which is dynamically recorded. A user-defined value will not function when a node is being created or modified.

Table 4-341 NodeMetadata

Parameter	Type	Description
name	String	<p>Node name</p> <p>NOTE</p> <p>Enter 1 to 56 characters starting with a letter and not ending with a hyphen (-). Only lowercase letters, digits, and hyphens (-) are allowed.</p> <p>If name is left empty or is not specified, the node name is generated based on the default rule, which is <i>Cluster name-Random characters</i>. Only the first 36 characters are kept for a long cluster name.</p> <p>If the number of nodes (count) is greater than 1, some random characters are added to the end of the entered node name according to the default rule, which is <i>User-defined name-Random characters</i>. Only the first 50 characters in the custom name are kept, followed by some random characters.</p> <p>Names of nodes running in clusters v1.28.1, v1.27.3, v1.25.6, v1.23.11, v1.21.12 or later support periods (.), but this configuration is not recommended.</p>
uid	String	<p>Node ID, which is unique and automatically generated after the resource is created. A user-defined ID will not take effect.</p>
labels	Map<String,String>	<p>CCE node label (not the native Kubernetes label).</p> <p>Labels are used to select objects that meet certain criteria. A label is a key-value pair.</p> <p>Example:</p> <pre>"labels": { "key" : "value" }</pre>
annotations	Map<String,String>	<p>CCE node annotations, in key-value pairs (not the native Kubernetes annotations).</p> <p>Example:</p> <pre>"annotations": { "key1" : "value1", "key2" : "value2" }</pre> <p>NOTE</p> <ul style="list-style-type: none"> annotations are not used to identify or select objects. The metadata in annotations may be small or large, structured or unstructured, and may include characters that are not allowed in labels. This parameter is used only for query and cannot be input through a request. Entered data of this parameter is invalid.
creationTimestamp	String	<p>Time when the object was created. The value is automatically generated after the object is created. A user-defined value will not take effect.</p>

Parameter	Type	Description
updateTimestamp	String	Time when the object was updated. The value is automatically generated after the object is created. A user-defined value will not take effect.

Table 4-342 NodeSpec

Parameter	Type	Description
flavor	String	Node specifications. For details about the node specifications supported by CCE, see Node Flavor Description .
az	String	AZ where the node to be created is located. You need to specify the AZ name. If it is set to random , the node will be created in a random AZ. For details about AZs supported by CCE, see Regions and Endpoints .
os	String	Node OS. For details about the supported OSs, see Node OS . NOTE <ul style="list-style-type: none"> The system automatically selects the supported OS based on the cluster version. If the current cluster version does not support the OS, an error will be reported. If alpha.cce/NodeImageID in extendParam is specified during node creation, you do not need to configure this parameter. This parameter is mandatory when creating a node pool. If you are using shared disk space when creating a node, this parameter is mandatory. Specifically, the disk initialization configuration management parameter storage is used and the name field of virtualSpaces in StorageGroups is set to share.
login	Login object	Node login mode. Either the key pair or password must be used for login.
rootVolume	Volume object	Information about disks on the node

Parameter	Type	Description
dataVolumes	Array of Volume objects	Data disk parameters of the node. Currently, you can add the second data disk for your node on the CCE console. This data disk is used by the container runtime and kubelet. Do not uninstall this disk. Otherwise, the node will become unavailable. For DeC nodes, the parameter description is the same as that for rootVolume .
storage	Storage object	Disk initialization management parameter. This parameter is complex to configure. For details, see Attaching Disks to a Node . If this parameter retains its default, disks are managed based on the DockerLVMConfigOverride (discarded) parameter in extendParam . This parameter is supported by clusters of version 1.15.11 and later. NOTE <ul style="list-style-type: none"> If a node specification involves both local disks and EVS disks, do not retain the default value of this parameter to prevent unexpected disk partitions. If you want to change the value range of a data disk to 20 to 32768, do not use the default value. If you want to use the shared disk space (with the runtime and Kubernetes partitions cancelled), do not retain the default value of this parameter. For details about the shared disk space, see Data Disk Space Allocation.
publicIP	NodePublicIP object	EIP of a node. NOTE This parameter is not supported when you add a node to a node pool.
nodeNicSpec	NodeNicSpec object	NIC of the node
count	Integer	Number of nodes to be created in a batch. The value must be a positive integer greater than or equal to 1 and less than or equal to the defined limit. This parameter can be left blank when it is used for a node pool.
billingMode	Integer	Node billing mode. <ul style="list-style-type: none"> 0: pay-per-use 1: yearly/monthly 2: (discarded) yearly/monthly billing with auto payment enabled

Parameter	Type	Description
taints	Array of Taint objects	<p>You can add taints to created nodes to configure anti-affinity. A maximum of 20 taints can be added. Each taint contains the following parameters:</p> <ul style="list-style-type: none"> • Key: A key must contain 1 to 63 characters starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain name can be used as the prefix of a key. • Value: A value must start with a letter or digit and can contain a maximum of 63 characters, including letters, digits, hyphens (-), underscores (_), and periods (.). • Effect: Available options are NoSchedule, PreferNoSchedule, and NoExecute. When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query. <p>Example:</p> <pre>"taints": [{ "key": "status", "value": "unavailable", "effect": "NoSchedule" }, { "key": "looks", "value": "bad", "effect": "NoSchedule" }]</pre>

Parameter	Type	Description
k8sTags	Map<String,String>	<p>Defined in key-value pairs. A maximum of 20 key-value pairs are allowed.</p> <ul style="list-style-type: none"> Key: Enter 1 to 63 characters, starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain can be prefixed to a key and contain a maximum of 253 characters. Example DNS subdomain: example.com/my-key Value: The value can be left blank or contain 1 to 63 characters that start with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed in the character string. When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query. <p>Example:</p> <pre>"k8sTags": { "key": "value" }</pre>
ecsGroupId	String	<p>ECS group ID. If this parameter is configured, nodes will be created in the specified ECS group.</p> <p>NOTE This configuration does not take effect when a node pool is created. To ensure that all nodes in a node pool are in the same ECS group, configure nodeManagement in the node pool.</p>
dedicatedHostId	String	<p>ID of the DeH to which the node is scheduled.</p> <p>NOTE This parameter is not supported when you add a node during node pool creation.</p>

Parameter	Type	Description
userTags	Array of UserTag objects	<p>Cloud server tag. The key of a tag must be unique. The maximum number of custom tags supported by CCE depends on the region and cannot exceed 8.</p> <p>When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query.</p> <p>NOTE Use only letters, Unicode characters, digits, special characters(-, _). Max characters: 36</p>
runtime	Runtime object	<p>Container runtime:</p> <ul style="list-style-type: none">• Clusters of v1.25 or earlier: docker.• Clusters of v1.25 or later: Container runtime varies with the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker. For nodes running other OSs, the default container runtime is containerd.

Parameter	Type	Description
initializedConditions	Array of strings	<p>Custom initialization flag, which is left blank by default.</p> <p>Before CCE nodes are initialized, node.cloudprovider.kubernetes.io/uninitialized is added to the nodes to prevent pods from being scheduled to these nodes. When creating a node, you can configure the initializedConditions parameter to control the taint removal time. By default, the timeout period is not configured.</p> <p>Example:</p> <ol style="list-style-type: none"> 1. Create a node and add parameter "initializedConditions": ["CCEInitial", "CustomedInitial"]. 2. After custom initialization is complete, CCE will call a Kubernetes API (for example, PATCH /v1/nodes/{node_ip}/status) to update the node conditions by adding two labels of types CCEInitial and CustomedInitial and setting the status to True, as shown in the following: <pre data-bbox="817 1055 1430 1205">status: conditions: - type: CCEInitial status: 'True' - type: CustomedInitial status: 'True'</pre> <ol style="list-style-type: none"> 1. CCE polls status.Conditions of nodes to check whether there are conditions of types CCEInitial and CustomedInitial. If such conditions exist and the status is True, the node initialization is complete and the initialization taint is removed. 2. initializedConditions allows you to configure a timeout period during node creation. For example, "initializedConditions": ["CCEInitial:15m", "CustomedInitial:15m"] indicates that the timeout period is 15 minutes. After the timeout period is reached, the initialization conditions will be automatically ignored and the initialization taint will be removed when CCE polls the node. <ul style="list-style-type: none"> ● Use only letters and digits. Max. characters: 20. ● Max. flags: 2. ● The unit of the timeout period is minute (m).

Parameter	Type	Description
extendParam	NodeExtendParam object	Extended parameters for creating a node.
hostnameConfig	HostnameConfig object	Kubernetes node name configuration parameter, which is supported by clusters of v1.23.6-r0 to v1.25 or clusters of v1.25.2-r0 or later versions.

Table 4-343 Login

Parameter	Type	Description
sshKey	String	Name of the key pair used for login.
userPassword	UserPassword object	Password used for node login.

Table 4-344 UserPassword

Parameter	Type	Description
username	String	Login account. The default value is root .
password	String	<p>If a username and a password are used to create a node, this field is shielded in the response body.</p> <p>A password must meet the following complexity requirements:</p> <ul style="list-style-type: none"> • Contains 8 to 26 characters. • Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (!@\$%^-_=+[{ } ; , / ? ~ # *) • Cannot contain the username or the username spelled backwards. <p>The password field must be salted during node creation. For details, see Adding a Salt in the password Field When Creating a Node.</p>

Table 4-345 Volume

Parameter	Type	Description
size	Integer	<p>Disk size, in GB.</p> <ul style="list-style-type: none"> • System disk: 40 to 1024 • Value range for the first data disk: 20 to 32768 (When storage is not specified, this value range becomes 100 to 32768.) • Value range for other data disks: 10 to 32768. (When storage is not specified, this value range becomes 100 to 32768.)
volumetype	String	<p>Disk type. For details about possible values, see the description of the root_volume parameter in the API used to create an ECS.</p> <ul style="list-style-type: none"> • SAS: high I/O SAS disk • SSD: ultra-high I/O SSD disk • SATA: common I/O SATA disk SATA disks have been removed from EVS. You can find them attached only on existing nodes.
extendParam	Map<String, Object>	<p>Extended disk parameters, defined in extendparam in the API used to create an ECS. For details, see Creating an ECS.</p>
cluster_id	String	<p>ID of the storage pool used by the ECS system disk. This field is used only for DeC clusters, which functions as dssPoolID, that is, the ID of the DSS storage pool.</p> <p>To obtain the value, see the ID field in the API used to obtain details of a DSS storage pool.</p>
cluster_type	String	<p>Storage class of the cloud server system disk. The value is always dss. This field is used only for DeC clusters.</p>
hw:passthrough	Boolean	<ul style="list-style-type: none"> • Pay attention to this field if your ECS is SDI-compliant. If the value of this field is true, an SCSI disk will be created. • If the node pool type is ElasticBMS, this field must be set to true. • If a node specification involves local disks and EVS disks at the same time, set the disk initialization parameters. For details, see Attaching Disks to a Node.
metadata	VolumeMetadata object	<p>EVS disk encryption information. This field is mandatory only when you need to encrypt the system disk or data disks of the node to be created.</p>

Table 4-346 VolumeMetadata

Parameter	Type	Description
<code>__system__encrypted</code>	String	Whether the EVS disk is encrypted. The value 0 indicates that the EVS disk is not encrypted, and the value 1 indicates that the EVS disk is encrypted. If this parameter is not specified, EVS disks will not be encrypted by default.
<code>__system__cmkid</code>	String	CMK ID, which indicates encryption in metadata . This field is used with __system__encrypted .

Table 4-347 Storage

Parameter	Type	Description
<code>storageSelectors</code>	Array of StorageSelectors objects	Disk selection. Matched disks are managed according to matchLabels and storageType .
<code>storageGroups</code>	Array of StorageGroups objects	A storage group consists of multiple storage devices. It is used to divide storage space.

Table 4-348 StorageSelectors

Parameter	Type	Description
<code>name</code>	String	Selector name, used as the index of selectorNames in storageGroup . Therefore, the name of each selector must be unique.
<code>storageType</code>	String	Specifies the storage type. Currently, only evs (EVS volumes) and local (local volumes) are supported. The local storage does not support disk selection. All local disks will form a VG. Therefore, only one storageSelector of the local type is allowed.
<code>matchLabels</code>	matchLabels object	Matching field of an EVS volume. The size , volumeType , metadataEncrypted , metadataCmkid and count fields are supported.

Table 4-349 matchLabels

Parameter	Type	Description
size	String	Matched disk size. If this parameter is left unspecified, the disk size is not limited. Example: 100
volumeType	String	EVS disk type. SSD, GPSSD, SAS, ESSD, and SATA are supported. If this parameter is left blank, there is no restriction on the disk type.
metadataEncrypted	String	Disk encryption identifier. If this parameter is set to 0 , it specifies that the disk is not encrypted. If this parameter is set to 1 , it specifies that the disk is encrypted. If this parameter is left blank, there is no restriction on the disk encryption identifier.
metadataCmkid	String	CMK ID of an encrypted disk. The value is a string of 36 bytes. If this parameter is left blank, there is no restriction on the disk key ID.
count	String	Number of disks to be selected. If this parameter is left blank, all disks of this type are selected.

Table 4-350 StorageGroups

Parameter	Type	Description
name	String	Name of a virtual storage group, which must be unique. NOTE <ul style="list-style-type: none"> If cceManaged is set to true, the name must be set to vgpaas. If the data disk is used as a temporary storage volume, the name must be vg-everest-localvolume-ephemeral. If the data disk is used as a persistent storage volume, the name must be vg-everest-localvolume-persistent.
cceManaged	Boolean	Storage space for Kubernetes and runtime components. Only one group can be set to true . If this parameter is left blank, the default value false is used.
selectorNames	Array of strings	This parameter corresponds to name in storageSelectors . A group can match multiple selectors, but a selector can match only one group.

Parameter	Type	Description
virtualSpaces	Array of VirtualSpace objects	Detailed management of space configuration in a group.

Table 4-351 VirtualSpace

Parameter	Type	Description
name	String	Name of a virtualSpace. Options: share , kubernetes , runtime , and user <ul style="list-style-type: none"> • share: shared space configuration (with the runtime and Kubernetes partitions cancelled). lvmConfig needs to be configured. • Kubernetes: Kubernetes space configuration. lvmConfig needs to be configured. • runtime: runtime space configuration. runtimeConfig needs to be configured. • user: user space configuration. lvmConfig needs to be configured.
size	String	Size of a virtualSpace. The value must be an integer in percentage. Example: 90%. NOTE The sum of the percentages of all virtualSpaces in a group cannot exceed 100%.
lvmConfig	LVMConfig object	LVM configuration managements, applicable to the <i>share</i> *, kubernetes , and user spaces. Note that one virtual space supports only one config.
runtimeConfig	RuntimeConfig object	runtime configurations, applicable to the runtime space. Note that one virtual space supports only one config.

Table 4-352 LVMConfig

Parameter	Type	Description
lvType	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.

Parameter	Type	Description
path	String	Path to which the disk is attached. This parameter takes effect only in user configuration. The value is an absolute path. Digits, letters, periods (.), hyphens (-), and underscores (_) are allowed.

Table 4-353 RuntimeConfig

Parameter	Type	Description
lvType	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.

Table 4-354 NodePublicIP

Parameter	Type	Description
ids	Array of strings	IDs of existing EIPs. The quantity cannot be greater than the number of nodes to be created. NOTE If ids has been set, you do not need to set count and eip .
count	Integer	Number of EIPs to be dynamically created. NOTE count and eip must be set at the same time.
eip	NodeEIPSpec object	EIP configuration.

Table 4-355 NodeEIPSpec

Parameter	Type	Description
iptype	String	EIP type, specified in publicip.type in the API for assigning an EIP. For details, see Assigning an EIP .
bandwidth	NodeBandwidth object	Bandwidth parameters of the EIP

Table 4-356 NodeBandwidth

Parameter	Type	Description
chargemode	String	<p>Bandwidth billing mode.</p> <ul style="list-style-type: none"> • If this field is not specified, the billing is based on bandwidth. • If the field is null, the billing is based on bandwidth. • If the field value is traffic, the billing is based on traffic. • If the value is out of the preceding options, the cloud server will fail to be created. <p>NOTE</p> <ul style="list-style-type: none"> • Billed by bandwidth: The billing will be based on the data transmission rate (in Mbps) of public networks. This billing mode is recommended if your bandwidth usage is higher than 10%. • Billed by traffic: The billing is based on the total amount of data (in GB) transmitted over the public network. This mode is available only when you are creating a pay-per-use node. This billing mode is recommended if your bandwidth usage is lower than 10%.
size	Integer	<p>Bandwidth size, specified in bandwidth.size in the API for assigning an EIP.</p> <p>For details, see Assigning an EIP.</p>
sharetype	String	<p>Bandwidth sharing type. Value options: PER (exclusive bandwidth)</p>

Table 4-357 NodeNicSpec

Parameter	Type	Description
primaryNic	NicSpec object	Description of the primary NIC.
extNics	Array of NicSpec objects	<p>Extension NIC</p> <p>NOTE This parameter is not supported when you add a node to a node pool.</p>

Table 4-358 NicSpec

Parameter	Type	Description
subnetId	String	Network ID of the subnet to which a NIC belongs. If subnetId is not specified when a primary NIC is creating, the cluster subnet will be used. If subnetList is also configured for a node pool, the subnetList field is used for adding subnets to the node pool. When creating an extension NIC, you must specify subnetId .
fixedIps	Array of strings	The IP address of the primary ENI is specified using fixedIps . The number of IP addresses cannot be greater than the number of created nodes. Either fixedIps or ipBlock can be specified. fixedIps cannot be specified for extension ENIs.
ipBlock	String	CIDR format of the primary NIC IP range. The IP address of the created node falls in this range. fixedIps and ipBlock cannot be specified at the same time.

Table 4-359 Taint

Parameter	Type	Description
key	String	Key.
value	String	Value.
effect	String	Effect.

Table 4-360 UserTag

Parameter	Type	Description
key	String	Key of the cloud server label. The value cannot start with CCE- or __type_baremetal .
value	String	Value of the cloud server label.

Table 4-361 Runtime

Parameter	Type	Description
name	String	<p>Container runtime. Default value:</p> <ul style="list-style-type: none"> • Clusters earlier than v1.25: The default value is docker. • Clusters of v1.25 or later: The default container runtime varies depending on the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker; for nodes running other OSs, the default container runtime is containerd.

Table 4-362 NodeExtendParam

Parameter	Type	Description
ecs:performancetype	String	ECS flavor types. This field is returned in the response.
orderId	String	Order ID. This parameter is returned in the response when the node is billed on a yearly/monthly basis with auto payment enabled (only in creation scenarios).
productId	String	Product ID. This field is returned in the response when the node is yearly/monthly-billed with auto payment enabled.
maxPods	Integer	<p>Maximum number of pods that can be created on a node, including the default system pods. Value range: 16 to 256.</p> <p>This limit prevents the node from being overloaded of pods.</p> <p>The number of pods that can be created on a node is determined by multiple parameters. For details, see Maximum Number of Pods That Can Be Created on a Node.</p>
periodType	String	<ul style="list-style-type: none"> • month: The unit is month. • year: The unit is year. <p>NOTE</p> <p>Request parameter, which is valid and mandatory only when billingMode is set to 1 (yearly/monthly billing) or 2 (yearly/monthly billing with auto renewal enabled) (discarded).</p> <p>Response parameter, which is returned only when a yearly/monthly node is created.</p>

Parameter	Type	Description
periodNum	Integer	<p>Subscription duration. The value can be:</p> <ul style="list-style-type: none"> • If periodType is month, the value ranges from 1 to 9. • If periodType is year, the value is 1. <p>NOTE Request parameter, which is valid and mandatory when billingMode is set to 1 or 2 (discarded). Response parameter, which is returned only when a yearly/monthly node is created.</p>
isAutoRenew	String	<p>Whether auto renewal is enabled.</p> <ul style="list-style-type: none"> • true: Auto renewal is enabled. • false: Auto renewal is not enabled. <p>NOTE This field is valid when billingMode is set to 1 or 2 (discarded). If not specified, auto renewal is not enabled.</p>
isAutoPay	String	<p>Whether to deduct fees automatically.</p> <ul style="list-style-type: none"> • true: Enable automatic fee deduction. • false: Do not enable automatic fee deduction. <p>NOTE This field is valid when billingMode is set to 1 or 2 (discarded). If not specified when billingMode is set to 1, auto fee deduction is not enabled. (Discarded: If not specified when billingMode is set to 2, auto fee deduction is enabled.)</p>

Parameter	Type	Description
DockerLVMConfigOverride	String	<p>Docker data disk configuration item. (This parameter has been discarded. Use the storage field instead.) Example default configuration: "DockerLVMConfigOverride":"dockerThinpool=vgpaas/90%VG;kubernetesLV=vgpaas/10%VG;diskType=evs;lvType=linear"</p> <p>By default, if no VD disk is available, an error occurs because the data disk fails to be found. Set diskType based on the actual drive letter type.</p> <p>The following fields are included:</p> <ul style="list-style-type: none"> • userLV (optional): size of the user space, for example, vgpaas/20%VG. • userPath (optional): mount path of the user space, for example, /home/wqt-test. • diskType: disk type. Currently, only evs, hdd, and ssd are supported. • lvType: type of a logic volume. The value can be linear or striped. • dockerThinpool: Docker space size, for example, vgpaas/60%VG. • kubernetesLV: kubelet space size, for example, vgpaas/20%VG.
dockerBaseSize	Integer	<p>Available disk space of a single container on a node, in GB. This parameter has been discarded. Use containerBaseSize.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In Device Mapper mode, the default value is 10. In OverlayFS mode, the available space of a single container is not limited by default, and the dockerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters of the new versions.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that dockerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device.</p>

Parameter	Type	Description
containerBaseSize	Integer	<p>Available disk space of a single container on a node, in GB.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In OverlayFS mode, the available space of a single container is not limited by default. In Device Mapper mode, the default value is 10, and the containerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters v1.23.14-r0, v1.25.9-r0, v1.27.6-r0, v1.28.4-r0 or later.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that containerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device. In new versions, the Device Mapper mode is used only by BMSs in the same resource pool and is being discarded.</p>
publicKey	String	Public key of a node.
alpha.cce/preInstall	String	<p>Pre-installation script.</p> <p>NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)</p>
alpha.cce/postInstall	String	<p>Post-installation script.</p> <p>NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)</p>
alpha.cce/NodeImageID	String	This parameter is required when a custom image is used to create a BMS node.

Parameter	Type	Description
nicMultiqueue	String	<ul style="list-style-type: none"> Number of ENI queues. Example setting: <code>"[{"queue":4}]"</code> The following fields are included: <ul style="list-style-type: none"> queue: number of ENI queues. This field can be configured only for BMS nodes in a CCE Turbo cluster. Supported proportions are {"1":128, "2":92, "4":92, "8":32, "16":16, "28":9}. That is, if there is one queue, a maximum of 128 ENIs can be bound. If there are two queues, a maximum of 92 ENIs can be bound for two queues. A larger number of ENI queues indicates higher performance but fewer ENIs can be bound. The queue settings cannot be changed after the node pool is created.
nicThreshold	String	<ul style="list-style-type: none"> ENI pre-binding thresholds. Example setting: <code>"0.3:0.6"</code> <ul style="list-style-type: none"> Low threshold (L): determines the minimum number of pre-bound ENIs (Min). Formula: $Min = Total\ number\ of\ ENIs\ of\ the\ node \times L$ High threshold (H): determines the maximum number of pre-bound ENIs (Max). Formula: $Max = Total\ number\ of\ ENIs\ of\ the\ node \times H$ Number of ENIs bound to a BMS node (B) and number of ENIs being used by pods (U): $U + Min < B < U + Max$ If the number of pre-bound ENIs on a BMS node is smaller than the minimum allowed, the system will bind more ENIs to make the numbers equal. If the number of pre-bound ENIs on a BMS node is larger than the maximum allowed, the system periodically unbinds ENIs (about every 2 minutes) to make the numbers equal. Both the thresholds are one-decimal-place values ranging from 0.0 to 1.0. The low threshold must be smaller than or equal to the high one. This field can be configured only for BMS nodes in a CCE Turbo cluster. Pre-binding ENIs can speed up workload creation but occupies IP addresses.

Parameter	Type	Description
chargingMode	Integer	Billing mode of a node. This parameter has been deprecated. Use the billingMode parameter in NodeSpec .
agency_name	String	Name of an agency An agency is created by a tenant administrator on Identity and Access Management (IAM) to provide temporary credentials for CCE nodes to access cloud servers. This parameter is returned only when it is transferred during node creation.
kubeReservedMem	Integer	Reserved node memory, which is reserved for Kubernetes components.
systemReservedMem	Integer	Reserved node memory, which is reserved for system components.
init-node-password	String	Node password. If this parameter is used as a response parameter, asterisks (*) will be displayed.

Table 4-363 HostnameConfig

Parameter	Type	Description
type	String	Configuration type of the Kubernetes node name. The default value is privatelp . <ul style="list-style-type: none"> • privatelp: The Kubernetes node is named after its IP address. • cceNodeName: The Kubernetes node is named after the CCE node. <p>NOTE</p> <ul style="list-style-type: none"> • For a node which is configured using cceNodeName, the name is the same as the Kubernetes node name and the ECS name. The node name cannot be changed. If the ECS name is changed on the ECS console, the node name will retain unchanged after ECS synchronization. • For a node which is configured using cceNodeName, to avoid a conflict between Kubernetes nodes, the system automatically adds a suffix to each node name. The suffix is in the format of A hyphen (-) Five random characters. The value of the random characters is a lowercase letter or a digit ranging from 0 to 9.

Table 4-364 NodeStatus

Parameter	Type	Description
phase	String	Node status, which is the status of a node during its lifecycle (such as installation and uninstallation) or the status of a node in a Kubernetes cluster. Options: <ul style="list-style-type: none"> ● Build: The node is being created. ● Installing: The node is being managed. ● Upgrading: The node is being upgraded. ● Active: The node is running properly. ● Abnormal: The node is abnormal. ● Deleting: The node is being deleted. ● Error: The node is faulty.
lastProbeTime	String	Last time when the node status was checked. If the cluster is in the abnormal, frozen, or intermediate state (for example, creating), the node status check may be affected. The node status that takes more than five minutes to check has no reference value.
jobID	String	ID of a creation or deletion job
serverId	String	ID of the underlying ECS or BMS node
privateIP	String	IP address in the private network segment of the primary NIC on the node
privateIPv6IP	String	IPv6 address in the private network segment of the primary NIC on the node
publicIP	String	Node EIP. If the ECS data is not synchronized in real time, you can click Sync Node Data on the console to manually update the data.
deleteStatus	DeleteStatus object	Resource status during resource deletion.

Table 4-365 DeleteStatus

Parameter	Type	Description
previous_total	Integer	Total number of existing cluster resource records when the cluster is deleted.
current_total	Integer	Latest number of resource records, which is generated based on the current cluster resource records.

Parameter	Type	Description
updated	Integer	Total number of resource records updated when the cluster is deleted.
added	Integer	Total number of resource records updated when the cluster is deleted.
deleted	Integer	Total number of resource records deleted when the cluster is deleted.

Example Requests

Update the name of a specified node.

```
{
  "metadata": {
    "name": "new-hostname"
  }
}
```

Example Responses

Status code: 200

Information about the specified node is successfully updated.

```
{
  "kind": "Node",
  "apiVersion": "v3",
  "metadata": {
    "name": "new-hostname",
    "uid": "4d1ecb2c-229a-11e8-9c75-0255ac100ceb",
    "creationTimestamp": "2017-08-20T21:11:09Z",
    "updateTimestamp": "2017-08-20T21:11:09Z",
    "annotations": {
      "kubernetes.io/node-pool.id": "az1.dc1#s1.medium#EulerOS 2.2"
    }
  },
  "spec": {
    "flavor": "s1.medium",
    "az": "az1.dc1",
    "os": "EulerOS 2.2",
    "login": {
      "sshKey": "KeyPair-001"
    },
    "rootVolume": {
      "volumetype": "SAS",
      "size": 40
    },
    "dataVolumes": [ {
      "volumetype": "SAS",
      "size": 100
    } ],
    "publicIP": {
      "eip": { }
    },
    "billingMode": 0
  },
  "status": {
    "phase": "Active",
    "serverId": "456789abc-9368-46f3-8f29-d1a95622a568",
    "publicIP": "10.34.56.78",
  }
}
```

```
"privateIP" : "192.168.1.23"  
}  
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Update the name of a specified node.

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cce.v3.region.CceRegion;  
import com.huaweicloud.sdk.cce.v3.*;  
import com.huaweicloud.sdk.cce.v3.model.*;  
  
public class UpdateNodeSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        CceClient client = CceClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))  
            .build();  
        UpdateNodeRequest request = new UpdateNodeRequest();  
        request.withClusterId("{cluster_id}");  
        request.withNodeId("{node_id}");  
        ClusterNodeInformation body = new ClusterNodeInformation();  
        ClusterNodeInformationMetadata metadatabody = new ClusterNodeInformationMetadata();  
        metadatabody.withName("new-hostname");  
        body.withMetadata(metadatabody);  
        request.withBody(body);  
        try {  
            UpdateNodeResponse response = client.updateNode(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

```
}  
}
```

Python

Update the name of a specified node.

```
# coding: utf-8  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkcce.v3.region.cce_region import CceRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkcce.v3 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = __import__('os').getenv("CLOUD_SDK_AK")  
    sk = __import__('os').getenv("CLOUD_SDK_SK")  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId) \  
  
    client = CceClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(CceRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = UpdateNodeRequest()  
        request.cluster_id = "{cluster_id}"  
        request.node_id = "{node_id}"  
        metadatabody = ClusterNodeInformationMetadata(  
            name="new-hostname"  
        )  
        request.body = ClusterNodeInformation(  
            metadata=metadatabody  
        )  
        response = client.update_node(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

Go

Update the name of a specified node.

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
```



```

example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := cce.NewCceClient(
    cce.CceClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateNodeRequest{}
request.ClusterId = "{cluster_id}"
request.NodeId = "{node_id}"
metadatabody := &model.ClusterNodeInformationMetadata{
    Name: "new-hostname",
}
request.Body = &model.ClusterNodeInformation{
    Metadata: metadatabody,
}
response, err := client.UpdateNode(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Information about the specified node is successfully updated.

Error Codes

See [Error Codes](#).

4.3.5 Deleting a Node

Function

This API is used to delete a specified node.

NOTE

The URL for cluster management is in the format of https://Endpoint/uri. In the URL, **uri** indicates the resource path, that is, the path for API access.

Calling Method

For details, see [Calling APIs](#).

URI

DELETE /api/v3/projects/{project_id}/clusters/{cluster_id}/nodes/{node_id}

Table 4-366 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
node_id	Yes	String	Node ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Table 4-367 Query Parameters

Parameter	Mandatory	Type	Description
nodepoolScaleDown	No	String	Whether the request is delivered by the node pool. If the value is not NoScaleDown , the number of pods in the corresponding node pool is automatically updated.

Request Parameters

Table 4-368 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-369 Response body parameters

Parameter	Type	Description
kind	String	API type. The value is fixed at Node and cannot be changed.
apiVersion	String	API version. The value is fixed at v3 and cannot be changed.
metadata	NodeMetadata object	Node metadata, which is a collection of attributes.
spec	NodeSpec object	Detailed description of the node. CCE creates or updates objects by defining or updating spec .
status	NodeStatus object	Node status, which is dynamically recorded. A user-defined value will not function when a node is being created or modified.

Table 4-370 NodeMetadata

Parameter	Type	Description
name	String	<p>Node name</p> <p>NOTE</p> <p>Enter 1 to 56 characters starting with a letter and not ending with a hyphen (-). Only lowercase letters, digits, and hyphens (-) are allowed.</p> <p>If name is left empty or is not specified, the node name is generated based on the default rule, which is <i>Cluster name-Random characters</i>. Only the first 36 characters are kept for a long cluster name.</p> <p>If the number of nodes (count) is greater than 1, some random characters are added to the end of the entered node name according to the default rule, which is <i>User-defined name-Random characters</i>. Only the first 50 characters in the custom name are kept, followed by some random characters.</p> <p>Names of nodes running in clusters v1.28.1, v1.27.3, v1.25.6, v1.23.11, v1.21.12 or later support periods (.), but this configuration is not recommended.</p>
uid	String	Node ID, which is unique and automatically generated after the resource is created. A user-defined ID will not take effect.
labels	Map<String,String>	<p>CCE node label (not the native Kubernetes label).</p> <p>Labels are used to select objects that meet certain criteria. A label is a key-value pair.</p> <p>Example:</p> <pre>"labels": { "key": "value" }</pre>
annotations	Map<String,String>	<p>CCE node annotations, in key-value pairs (not the native Kubernetes annotations).</p> <p>Example:</p> <pre>"annotations": { "key1": "value1", "key2": "value2" }</pre> <p>NOTE</p> <ul style="list-style-type: none"> annotations are not used to identify or select objects. The metadata in annotations may be small or large, structured or unstructured, and may include characters that are not allowed in labels. This parameter is used only for query and cannot be input through a request. Entered data of this parameter is invalid.
creationTimestamp	String	Time when the object was created. The value is automatically generated after the object is created. A user-defined value will not take effect.

Parameter	Type	Description
updateTimestamp	String	Time when the object was updated. The value is automatically generated after the object is created. A user-defined value will not take effect.

Table 4-371 NodeSpec

Parameter	Type	Description
flavor	String	Node specifications. For details about the node specifications supported by CCE, see Node Flavor Description .
az	String	AZ where the node to be created is located. You need to specify the AZ name. If it is set to random , the node will be created in a random AZ. For details about AZs supported by CCE, see Regions and Endpoints .
os	String	Node OS. For details about the supported OSs, see Node OS . NOTE <ul style="list-style-type: none"> The system automatically selects the supported OS based on the cluster version. If the current cluster version does not support the OS, an error will be reported. If alpha.cce/NodeImageID in extendParam is specified during node creation, you do not need to configure this parameter. This parameter is mandatory when creating a node pool. If you are using shared disk space when creating a node, this parameter is mandatory. Specifically, the disk initialization configuration management parameter storage is used and the name field of virtualSpaces in StorageGroups is set to share.
login	Login object	Node login mode. Either the key pair or password must be used for login.
rootVolume	Volume object	Information about disks on the node

Parameter	Type	Description
dataVolumes	Array of Volume objects	Data disk parameters of the node. Currently, you can add the second data disk for your node on the CCE console. This data disk is used by the container runtime and kubelet. Do not uninstall this disk. Otherwise, the node will become unavailable. For DeC nodes, the parameter description is the same as that for rootVolume .
storage	Storage object	Disk initialization management parameter. This parameter is complex to configure. For details, see Attaching Disks to a Node . If this parameter retains its default, disks are managed based on the DockerLVMConfigOverride (discarded) parameter in extendParam . This parameter is supported by clusters of version 1.15.11 and later. NOTE <ul style="list-style-type: none"> If a node specification involves both local disks and EVS disks, do not retain the default value of this parameter to prevent unexpected disk partitions. If you want to change the value range of a data disk to 20 to 32768, do not use the default value. If you want to use the shared disk space (with the runtime and Kubernetes partitions cancelled), do not retain the default value of this parameter. For details about the shared disk space, see Data Disk Space Allocation.
publicIP	NodePublicIP object	EIP of a node. NOTE This parameter is not supported when you add a node to a node pool.
nodeNicSpec	NodeNicSpec object	NIC of the node
count	Integer	Number of nodes to be created in a batch. The value must be a positive integer greater than or equal to 1 and less than or equal to the defined limit. This parameter can be left blank when it is used for a node pool.
billingMode	Integer	Node billing mode. <ul style="list-style-type: none"> 0: pay-per-use 1: yearly/monthly 2: (discarded) yearly/monthly billing with auto payment enabled

Parameter	Type	Description
taints	Array of Taint objects	<p>You can add taints to created nodes to configure anti-affinity. A maximum of 20 taints can be added. Each taint contains the following parameters:</p> <ul style="list-style-type: none"> • Key: A key must contain 1 to 63 characters starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain name can be used as the prefix of a key. • Value: A value must start with a letter or digit and can contain a maximum of 63 characters, including letters, digits, hyphens (-), underscores (_), and periods (.). • Effect: Available options are NoSchedule, PreferNoSchedule, and NoExecute. When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query. <p>Example:</p> <pre> taints: [{ "key": "status", "value": "unavailable", "effect": "NoSchedule" }, { "key": "looks", "value": "bad", "effect": "NoSchedule" }] </pre>

Parameter	Type	Description
k8sTags	Map<String,String>	<p>Defined in key-value pairs. A maximum of 20 key-value pairs are allowed.</p> <ul style="list-style-type: none"> Key: Enter 1 to 63 characters, starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain can be prefixed to a key and contain a maximum of 253 characters. Example DNS subdomain: example.com/my-key Value: The value can be left blank or contain 1 to 63 characters that start with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed in the character string. When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query. <p>Example:</p> <pre>"k8sTags": { "key": "value" }</pre>
ecsGroupId	String	<p>ECS group ID. If this parameter is configured, nodes will be created in the specified ECS group.</p> <p>NOTE This configuration does not take effect when a node pool is created. To ensure that all nodes in a node pool are in the same ECS group, configure nodeManagement in the node pool.</p>
dedicatedHostId	String	<p>ID of the DeH to which the node is scheduled.</p> <p>NOTE This parameter is not supported when you add a node during node pool creation.</p>

Parameter	Type	Description
userTags	Array of UserTag objects	<p>Cloud server tag. The key of a tag must be unique. The maximum number of custom tags supported by CCE depends on the region and cannot exceed 8.</p> <p>When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query.</p> <p>NOTE Use only letters, Unicode characters, digits, special characters(-, _). Max characters: 36</p>
runtime	Runtime object	<p>Container runtime:</p> <ul style="list-style-type: none"> • Clusters of v1.25 or earlier: docker. • Clusters of v1.25 or later: Container runtime varies with the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker. For nodes running other OSs, the default container runtime is containerd.

Parameter	Type	Description
initializedConditions	Array of strings	<p>Custom initialization flag, which is left blank by default.</p> <p>Before CCE nodes are initialized, node.cloudprovider.kubernetes.io/uninitialized is added to the nodes to prevent pods from being scheduled to these nodes. When creating a node, you can configure the initializedConditions parameter to control the taint removal time. By default, the timeout period is not configured.</p> <p>Example:</p> <ol style="list-style-type: none"> 1. Create a node and add parameter "initializedConditions": ["CCEInitial", "CustomedInitial"]. 2. After custom initialization is complete, CCE will call a Kubernetes API (for example, PATCH /v1/nodes/{node_ip}/status) to update the node conditions by adding two labels of types CCEInitial and CustomedInitial and setting the status to True, as shown in the following: <pre data-bbox="815 1055 1430 1205">status: conditions: - type: CCEInitial status: 'True' - type: CustomedInitial status: 'True'</pre> <ol style="list-style-type: none"> 1. CCE polls status.Conditions of nodes to check whether there are conditions of types CCEInitial and CustomedInitial. If such conditions exist and the status is True, the node initialization is complete and the initialization taint is removed. 2. initializedConditions allows you to configure a timeout period during node creation. For example, "initializedConditions": ["CCEInitial:15m", "CustomedInitial:15m"] indicates that the timeout period is 15 minutes. After the timeout period is reached, the initialization conditions will be automatically ignored and the initialization taint will be removed when CCE polls the node. <ul style="list-style-type: none"> ● Use only letters and digits. Max. characters: 20. ● Max. flags: 2. ● The unit of the timeout period is minute (m).

Parameter	Type	Description
extendParam	NodeExtendParam object	Extended parameters for creating a node.
hostnameConfig	HostnameConfig object	Kubernetes node name configuration parameter, which is supported by clusters of v1.23.6-r0 to v1.25 or clusters of v1.25.2-r0 or later versions.

Table 4-372 Login

Parameter	Type	Description
sshKey	String	Name of the key pair used for login.
userPassword	UserPassword object	Password used for node login.

Table 4-373 UserPassword

Parameter	Type	Description
username	String	Login account. The default value is root .
password	String	<p>If a username and a password are used to create a node, this field is shielded in the response body.</p> <p>A password must meet the following complexity requirements:</p> <ul style="list-style-type: none"> • Contains 8 to 26 characters. • Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (!@\$%^-_=+[{ }];,./?~#*) • Cannot contain the username or the username spelled backwards. <p>The password field must be salted during node creation. For details, see Adding a Salt in the password Field When Creating a Node.</p>

Table 4-374 Volume

Parameter	Type	Description
size	Integer	Disk size, in GB. <ul style="list-style-type: none"> System disk: 40 to 1024 Value range for the first data disk: 20 to 32768 (When storage is not specified, this value range becomes 100 to 32768.) Value range for other data disks: 10 to 32768. (When storage is not specified, this value range becomes 100 to 32768.)
volumetype	String	Disk type. For details about possible values, see the description of the root_volume parameter in the API used to create an ECS. <ul style="list-style-type: none"> SAS: high I/O SAS disk SSD: ultra-high I/O SSD disk SATA: common I/O SATA disk SATA disks have been removed from EVS. You can find them attached only on existing nodes.
extendParam	Map<String, Object>	Extended disk parameters, defined in extendparam in the API used to create an ECS. For details, see Creating an ECS .
cluster_id	String	ID of the storage pool used by the ECS system disk. This field is used only for DeC clusters, which functions as dssPoolID , that is, the ID of the DSS storage pool. To obtain the value, see the ID field in the API used to obtain details of a DSS storage pool .
cluster_type	String	Storage class of the cloud server system disk. The value is always dss . This field is used only for DeC clusters.
hw:passthrough	Boolean	<ul style="list-style-type: none"> Pay attention to this field if your ECS is SDI-compliant. If the value of this field is true, an SCSI disk will be created. If the node pool type is ElasticBMS, this field must be set to true. If a node specification involves local disks and EVS disks at the same time, set the disk initialization parameters. For details, see Attaching Disks to a Node.
metadata	VolumeMetadata object	EVS disk encryption information. This field is mandatory only when you need to encrypt the system disk or data disks of the node to be created.

Table 4-375 VolumeMetadata

Parameter	Type	Description
<code>__system__encrypted</code>	String	Whether the EVS disk is encrypted. The value 0 indicates that the EVS disk is not encrypted, and the value 1 indicates that the EVS disk is encrypted. If this parameter is not specified, EVS disks will not be encrypted by default.
<code>__system__cmkid</code>	String	CMK ID, which indicates encryption in metadata . This field is used with __system__encrypted .

Table 4-376 Storage

Parameter	Type	Description
<code>storageSelectors</code>	Array of StorageSelectors objects	Disk selection. Matched disks are managed according to matchLabels and storageType .
<code>storageGroups</code>	Array of StorageGroups objects	A storage group consists of multiple storage devices. It is used to divide storage space.

Table 4-377 StorageSelectors

Parameter	Type	Description
<code>name</code>	String	Selector name, used as the index of selectorNames in storageGroup . Therefore, the name of each selector must be unique.
<code>storageType</code>	String	Specifies the storage type. Currently, only evs (EVS volumes) and local (local volumes) are supported. The local storage does not support disk selection. All local disks will form a VG. Therefore, only one storageSelector of the local type is allowed.
<code>matchLabels</code>	matchLabels object	Matching field of an EVS volume. The size , volumeType , metadataEncrypted , metadataCmkid and count fields are supported.

Table 4-378 matchLabels

Parameter	Type	Description
size	String	Matched disk size. If this parameter is left unspecified, the disk size is not limited. Example: 100
volumeType	String	EVS disk type. SSD, GPSSD, SAS, ESSD, and SATA are supported. If this parameter is left blank, there is no restriction on the disk type.
metadataEncrypted	String	Disk encryption identifier. If this parameter is set to 0 , it specifies that the disk is not encrypted. If this parameter is set to 1 , it specifies that the disk is encrypted. If this parameter is left blank, there is no restriction on the disk encryption identifier.
metadataCmkid	String	CMK ID of an encrypted disk. The value is a string of 36 bytes. If this parameter is left blank, there is no restriction on the disk key ID.
count	String	Number of disks to be selected. If this parameter is left blank, all disks of this type are selected.

Table 4-379 StorageGroups

Parameter	Type	Description
name	String	Name of a virtual storage group, which must be unique. NOTE <ul style="list-style-type: none"> If cceManaged is set to true, the name must be set to vgpaas. If the data disk is used as a temporary storage volume, the name must be vg-everest-localvolume-ephemeral. If the data disk is used as a persistent storage volume, the name must be vg-everest-localvolume-persistent.
cceManaged	Boolean	Storage space for Kubernetes and runtime components. Only one group can be set to true . If this parameter is left blank, the default value false is used.
selectorNames	Array of strings	This parameter corresponds to name in storageSelectors . A group can match multiple selectors, but a selector can match only one group.

Parameter	Type	Description
virtualSpaces	Array of VirtualSpace objects	Detailed management of space configuration in a group.

Table 4-380 VirtualSpace

Parameter	Type	Description
name	String	Name of a virtualSpace. Options: share , kubernetes , runtime , and user <ul style="list-style-type: none"> • share: shared space configuration (with the runtime and Kubernetes partitions cancelled). lvmConfig needs to be configured. • Kubernetes: Kubernetes space configuration. lvmConfig needs to be configured. • runtime: runtime space configuration. runtimeConfig needs to be configured. • user: user space configuration. lvmConfig needs to be configured.
size	String	Size of a virtualSpace. The value must be an integer in percentage. Example: 90%. NOTE The sum of the percentages of all virtualSpaces in a group cannot exceed 100%.
lvmConfig	LVMConfig object	LVM configuration managements, applicable to the <i>share</i> *, kubernetes , and user spaces. Note that one virtual space supports only one config.
runtimeConfig	RuntimeConfig object	runtime configurations, applicable to the runtime space. Note that one virtual space supports only one config.

Table 4-381 LVMConfig

Parameter	Type	Description
lvType	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.

Parameter	Type	Description
path	String	Path to which the disk is attached. This parameter takes effect only in user configuration. The value is an absolute path. Digits, letters, periods (.), hyphens (-), and underscores (_) are allowed.

Table 4-382 RuntimeConfig

Parameter	Type	Description
lvType	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.

Table 4-383 NodePublicIP

Parameter	Type	Description
ids	Array of strings	IDs of existing EIPs. The quantity cannot be greater than the number of nodes to be created. NOTE If ids has been set, you do not need to set count and eip .
count	Integer	Number of EIPs to be dynamically created. NOTE count and eip must be set at the same time.
eip	NodeEIPSpec object	EIP configuration.

Table 4-384 NodeEIPSpec

Parameter	Type	Description
iptype	String	EIP type, specified in publicip.type in the API for assigning an EIP. For details, see Assigning an EIP .
bandwidth	NodeBandwidth object	Bandwidth parameters of the EIP

Table 4-385 NodeBandwidth

Parameter	Type	Description
chargemode	String	<p>Bandwidth billing mode.</p> <ul style="list-style-type: none"> • If this field is not specified, the billing is based on bandwidth. • If the field is null, the billing is based on bandwidth. • If the field value is traffic, the billing is based on traffic. • If the value is out of the preceding options, the cloud server will fail to be created. <p>NOTE</p> <ul style="list-style-type: none"> • Billed by bandwidth: The billing will be based on the data transmission rate (in Mbps) of public networks. This billing mode is recommended if your bandwidth usage is higher than 10%. • Billed by traffic: The billing is based on the total amount of data (in GB) transmitted over the public network. This mode is available only when you are creating a pay-per-use node. This billing mode is recommended if your bandwidth usage is lower than 10%.
size	Integer	<p>Bandwidth size, specified in bandwidth.size in the API for assigning an EIP.</p> <p>For details, see Assigning an EIP.</p>
sharetype	String	<p>Bandwidth sharing type. Value options: PER (exclusive bandwidth)</p>

Table 4-386 NodeNicSpec

Parameter	Type	Description
primaryNic	NicSpec object	Description of the primary NIC.
extNics	Array of NicSpec objects	<p>Extension NIC</p> <p>NOTE This parameter is not supported when you add a node to a node pool.</p>

Table 4-387 NicSpec

Parameter	Type	Description
subnetId	String	Network ID of the subnet to which a NIC belongs. If subnetId is not specified when a primary NIC is creating, the cluster subnet will be used. If subnetList is also configured for a node pool, the subnetList field is used for adding subnets to the node pool. When creating an extension NIC, you must specify subnetId .
fixedIps	Array of strings	The IP address of the primary ENI is specified using fixedIps . The number of IP addresses cannot be greater than the number of created nodes. Either fixedIps or ipBlock can be specified. fixedIps cannot be specified for extension ENIs.
ipBlock	String	CIDR format of the primary NIC IP range. The IP address of the created node falls in this range. fixedIps and ipBlock cannot be specified at the same time.

Table 4-388 Taint

Parameter	Type	Description
key	String	Key.
value	String	Value.
effect	String	Effect.

Table 4-389 UserTag

Parameter	Type	Description
key	String	Key of the cloud server label. The value cannot start with CCE- or __type_baremetal .
value	String	Value of the cloud server label.

Table 4-390 Runtime

Parameter	Type	Description
name	String	<p>Container runtime. Default value:</p> <ul style="list-style-type: none"> • Clusters earlier than v1.25: The default value is docker. • Clusters of v1.25 or later: The default container runtime varies depending on the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker; for nodes running other OSs, the default container runtime is containerd.

Table 4-391 NodeExtendParam

Parameter	Type	Description
ecs:performancetype	String	ECS flavor types. This field is returned in the response.
orderId	String	Order ID. This parameter is returned in the response when the node is billed on a yearly/monthly basis with auto payment enabled (only in creation scenarios).
productId	String	Product ID. This field is returned in the response when the node is yearly/monthly-billed with auto payment enabled.
maxPods	Integer	<p>Maximum number of pods that can be created on a node, including the default system pods. Value range: 16 to 256.</p> <p>This limit prevents the node from being overloaded of pods.</p> <p>The number of pods that can be created on a node is determined by multiple parameters. For details, see Maximum Number of Pods That Can Be Created on a Node.</p>
periodType	String	<ul style="list-style-type: none"> • month: The unit is month. • year: The unit is year. <p>NOTE</p> <p>Request parameter, which is valid and mandatory only when billingMode is set to 1 (yearly/monthly billing) or 2 (yearly/monthly billing with auto renewal enabled) (discarded).</p> <p>Response parameter, which is returned only when a yearly/monthly node is created.</p>

Parameter	Type	Description
periodNum	Integer	<p>Subscription duration. The value can be:</p> <ul style="list-style-type: none"> • If periodType is month, the value ranges from 1 to 9. • If periodType is year, the value is 1. <p>NOTE Request parameter, which is valid and mandatory when billingMode is set to 1 or 2 (discarded). Response parameter, which is returned only when a yearly/monthly node is created.</p>
isAutoRenew	String	<p>Whether auto renewal is enabled.</p> <ul style="list-style-type: none"> • true: Auto renewal is enabled. • false: Auto renewal is not enabled. <p>NOTE This field is valid when billingMode is set to 1 or 2 (discarded). If not specified, auto renewal is not enabled.</p>
isAutoPay	String	<p>Whether to deduct fees automatically.</p> <ul style="list-style-type: none"> • true: Enable automatic fee deduction. • false: Do not enable automatic fee deduction. <p>NOTE This field is valid when billingMode is set to 1 or 2 (discarded). If not specified when billingMode is set to 1, auto fee deduction is not enabled. (Discarded: If not specified when billingMode is set to 2, auto fee deduction is enabled.)</p>

Parameter	Type	Description
DockerLVMConfigOverride	String	<p>Docker data disk configuration item. (This parameter has been discarded. Use the storage field instead.) Example default configuration: <code>"DockerLVMConfigOverride":"dockerThinpool=vgpaas/90%VG;kubernetesLV=vgpaas/10%VG;diskType=evs;lvType=linear"</code></p> <p>By default, if no VD disk is available, an error occurs because the data disk fails to be found. Set diskType based on the actual drive letter type.</p> <p>The following fields are included:</p> <ul style="list-style-type: none"> • userLV (optional): size of the user space, for example, vgpaas/20%VG. • userPath (optional): mount path of the user space, for example, /home/wqt-test. • diskType: disk type. Currently, only evs, hdd, and ssd are supported. • lvType: type of a logic volume. The value can be linear or striped. • dockerThinpool: Docker space size, for example, vgpaas/60%VG. • kubernetesLV: kubelet space size, for example, vgpaas/20%VG.
dockerBaseSize	Integer	<p>Available disk space of a single container on a node, in GB. This parameter has been discarded. Use containerBaseSize.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In Device Mapper mode, the default value is 10. In OverlayFS mode, the available space of a single container is not limited by default, and the dockerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters of the new versions.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that dockerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device.</p>

Parameter	Type	Description
containerBaseSize	Integer	<p>Available disk space of a single container on a node, in GB.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In OverlayFS mode, the available space of a single container is not limited by default. In Device Mapper mode, the default value is 10, and the containerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters v1.23.14-r0, v1.25.9-r0, v1.27.6-r0, v1.28.4-r0 or later.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that containerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device. In new versions, the Device Mapper mode is used only by BMSs in the same resource pool and is being discarded.</p>
publicKey	String	Public key of a node.
alpha.cce/preInstall	String	<p>Pre-installation script.</p> <p>NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)</p>
alpha.cce/postInstall	String	<p>Post-installation script.</p> <p>NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)</p>
alpha.cce/NodeImageID	String	This parameter is required when a custom image is used to create a BMS node.

Parameter	Type	Description
nicMultiqueue	String	<ul style="list-style-type: none"> Number of ENI queues. Example setting: <code>"[{"queue":4}]"</code> The following fields are included: <ul style="list-style-type: none"> queue: number of ENI queues. This field can be configured only for BMS nodes in a CCE Turbo cluster. Supported proportions are {"1":128, "2":92, "4":92, "8":32, "16":16, "28":9}. That is, if there is one queue, a maximum of 128 ENIs can be bound. If there are two queues, a maximum of 92 ENIs can be bound for two queues. A larger number of ENI queues indicates higher performance but fewer ENIs can be bound. The queue settings cannot be changed after the node pool is created.
nicThreshold	String	<ul style="list-style-type: none"> ENI pre-binding thresholds. Example setting: <code>"0.3:0.6"</code> <ul style="list-style-type: none"> Low threshold (L): determines the minimum number of pre-bound ENIs (Min). Formula: $Min = Total\ number\ of\ ENIs\ of\ the\ node \times L$ High threshold (H): determines the maximum number of pre-bound ENIs (Max). Formula: $Max = Total\ number\ of\ ENIs\ of\ the\ node \times H$ Number of ENIs bound to a BMS node (B) and number of ENIs being used by pods (U): $U + Min < B < U + Max$ If the number of pre-bound ENIs on a BMS node is smaller than the minimum allowed, the system will bind more ENIs to make the numbers equal. If the number of pre-bound ENIs on a BMS node is larger than the maximum allowed, the system periodically unbinds ENIs (about every 2 minutes) to make the numbers equal. Both the thresholds are one-decimal-place values ranging from 0.0 to 1.0. The low threshold must be smaller than or equal to the high one. This field can be configured only for BMS nodes in a CCE Turbo cluster. Pre-binding ENIs can speed up workload creation but occupies IP addresses.

Parameter	Type	Description
chargingMode	Integer	Billing mode of a node. This parameter has been deprecated. Use the billingMode parameter in NodeSpec .
agency_name	String	Name of an agency An agency is created by a tenant administrator on Identity and Access Management (IAM) to provide temporary credentials for CCE nodes to access cloud servers. This parameter is returned only when it is transferred during node creation.
kubeReservedMem	Integer	Reserved node memory, which is reserved for Kubernetes components.
systemReservedMem	Integer	Reserved node memory, which is reserved for system components.
init-node-password	String	Node password. If this parameter is used as a response parameter, asterisks (*) will be displayed.

Table 4-392 HostnameConfig

Parameter	Type	Description
type	String	Configuration type of the Kubernetes node name. The default value is privatelp . <ul style="list-style-type: none"> • privatelp: The Kubernetes node is named after its IP address. • cceNodeName: The Kubernetes node is named after the CCE node. <p>NOTE</p> <ul style="list-style-type: none"> • For a node which is configured using cceNodeName, the name is the same as the Kubernetes node name and the ECS name. The node name cannot be changed. If the ECS name is changed on the ECS console, the node name will retain unchanged after ECS synchronization. • For a node which is configured using cceNodeName, to avoid a conflict between Kubernetes nodes, the system automatically adds a suffix to each node name. The suffix is in the format of A hyphen (-) Five random characters. The value of the random characters is a lowercase letter or a digit ranging from 0 to 9.

Table 4-393 NodeStatus

Parameter	Type	Description
phase	String	Node status, which is the status of a node during its lifecycle (such as installation and uninstallation) or the status of a node in a Kubernetes cluster. Options: <ul style="list-style-type: none"> ● Build: The node is being created. ● Installing: The node is being managed. ● Upgrading: The node is being upgraded. ● Active: The node is running properly. ● Abnormal: The node is abnormal. ● Deleting: The node is being deleted. ● Error: The node is faulty.
lastProbeTime	String	Last time when the node status was checked. If the cluster is in the abnormal, frozen, or intermediate state (for example, creating), the node status check may be affected. The node status that takes more than five minutes to check has no reference value.
jobID	String	ID of a creation or deletion job
serverId	String	ID of the underlying ECS or BMS node
privateIP	String	IP address in the private network segment of the primary NIC on the node
privateIPv6IP	String	IPv6 address in the private network segment of the primary NIC on the node
publicIP	String	Node EIP. If the ECS data is not synchronized in real time, you can click Sync Node Data on the console to manually update the data.
deleteStatus	DeleteStatus object	Resource status during resource deletion.

Table 4-394 DeleteStatus

Parameter	Type	Description
previous_total	Integer	Total number of existing cluster resource records when the cluster is deleted.
current_total	Integer	Latest number of resource records, which is generated based on the current cluster resource records.

Parameter	Type	Description
updated	Integer	Total number of resource records updated when the cluster is deleted.
added	Integer	Total number of resource records updated when the cluster is deleted.
deleted	Integer	Total number of resource records deleted when the cluster is deleted.

Example Requests

None

Example Responses

Status code: 200

The job for deleting a node is successfully delivered.

```
{
  "kind": "Node",
  "apiVersion": "v3",
  "metadata": {
    "name": "new-hostname",
    "uid": "cc697ad9-9563-11e8-8ea7-0255ac106311",
    "creationTimestamp": "2018-08-01 08:20:49.944664515 +0000 UTC",
    "updateTimestamp": "2018-08-01 09:20:05.644032347 +0000 UTC",
    "annotations": {
      "kubernetes.io/node-pool.id": "az1.dc1#s1.medium#EulerOS 2.2"
    }
  },
  "spec": {
    "flavor": "s1.medium",
    "az": "az1.dc1",
    "os": "EulerOS 2.2",
    "login": {
      "sshKey": "KeyPair-001"
    },
    "rootVolume": {
      "volumetype": "SAS",
      "size": 40
    },
    "dataVolumes": [ {
      "volumetype": "SAS",
      "size": 100
    } ],
    "publicIP": {
      "eip": {
        "bandwidth": { }
      }
    },
    "billingMode": 0
  },
  "status": {
    "phase": "Active",
    "jobID": "661f6f7d-956c-11e8-a916-0255ac10575d",
    "serverId": "5b504f8d-33f1-4ab7-a600-b62dac967d72",
    "privateIP": "192.168.0.69",
    "publicIP": "10.154.194.59"
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class DeleteNodeSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteNodeRequest request = new DeleteNodeRequest();
        request.withClusterId("{cluster_id}");
        request.withNodeId("{node_id}");
        try {
            DeleteNodeResponse response = client.deleteNode(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *
```

```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteNodeRequest()
        request.cluster_id = "{cluster_id}"
        request.node_id = "{node_id}"
        response = client.delete_node(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteNodeRequest{}
    request.ClusterId = "{cluster_id}"
    request.NodeId = "{node_id}"
    response, err := client.DeleteNode(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    }
}
```

```

    } else {
      fmt.Println(err)
    }
  }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The job for deleting a node is successfully delivered.

Error Codes

See [Error Codes](#).

4.3.6 Accepting a Node

Function

This API is used to accept a node into a specified cluster.

NOTE

The URL for cluster management is in the format of https://Endpoint/uri. In the URL, **uri** indicates the resource path, that is, the path for API access.

Calling Method

For details, see [Calling APIs](#).

URI

POST /api/v3/projects/{project_id}/clusters/{cluster_id}/nodes/add

Table 4-395 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Parameter	Mandatory	Type	Description
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-396 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Table 4-397 Request body parameters

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	API version. The value is fixed at v3 .
kind	Yes	String	API type. The value is fixed at List .
nodeList	Yes	Array of AddNode objects	List of nodes to be managed. A maximum of 200 nodes can be managed simultaneously.

Table 4-398 AddNode

Parameter	Mandatory	Type	Description
serverID	Yes	String	Server ID. For details about how to obtain the server ID, see the ECS or BMS documentation.

Parameter	Mandatory	Type	Description
spec	Yes	ReinstallNodeSpec object	Node reinstallation configuration parameters. Currently, accepted nodes cannot be added into node pools.

Table 4-399 ReinstallNodeSpec

Parameter	Mandatory	Type	Description
os	Yes	String	Operating system. If you specify a custom image, the actual OS version in the IMS image is used. Select an OS version supported by the current cluster, for example, EulerOS 2.5, CentOS 7.6, or EulerOS 2.8.
login	Yes	Login object	Node login mode. Either the key pair or password must be used for login.
name	No	String	Node name. NOTE Specifying this field during reinstallation will change the node name, and the server name will change accordingly. By default, the current server name is used as the node name. Enter 1 to 56 characters starting with a letter and not ending with a hyphen (-).
serverConfig	No	ReinstallServerConfig object	Server configuration.
volumeConfig	No	ReinstallVolumeConfig object	Volume management configuration.
runtimeConfig	No	ReinstallRuntimeConfig object	Container runtime configuration.
k8sOptions	No	ReinstallK8sOptionsConfig object	Kubernetes node configuration.

Parameter	Mandatory	Type	Description
lifecycle	No	NodeLifecycleConfig object	Customized lifecycle configuration of a node.
initializedConditions	No	Array of strings	<p>Custom initialization flag. Before CCE nodes are initialized, they are tainted with node.cloudprovider.kubernetes.io/uninitialized to prevent pods from being scheduled to them.</p> <p>CCE supports custom initialization flags. After receiving the initializedConditions parameter, CCE converts the parameter value into a node label and provisions the label with the node, for example, cloudprovider.openvessel.io/inject-initialized-conditions=CCEInitial_CustomedInitial.</p> <p>After the node is labeled, its status.Conditions is polled to check whether the type of conditions has a flag name, such as CCEInitial and CustomedInitial. If all input flags exist and their status is True, the node initialization is complete and the initialization taint is removed.</p> <ul style="list-style-type: none"> • Use only letters and digits. Max. characters: 20. • Max. flags: 2.
extendParam	No	ReinstallExtendParam object	Extended reinstallation parameter, which is discarded.
hostnameConfig	No	HostnameConfig object	Kubernetes node name configuration parameter, which is supported by clusters of v1.23.6-r0 to v1.25 or clusters of v1.25.2-r0 or later versions.

Table 4-400 Login

Parameter	Mandatory	Type	Description
sshKey	No	String	Name of the key pair used for login.
userPassword	No	UserPassword object	Password used for node login.

Table 4-401 UserPassword

Parameter	Mandatory	Type	Description
username	No	String	Login account. The default value is root .
password	Yes	String	<p>If a username and a password are used to create a node, this field is shielded in the response body.</p> <p>A password must meet the following complexity requirements:</p> <ul style="list-style-type: none"> • Contains 8 to 26 characters. • Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (!@\$%^-_=+[{ }];,./?~#*) • Cannot contain the username or the username spelled backwards. <p>The password field must be salted during node creation. For details, see Adding a Salt in the password Field When Creating a Node.</p>

Table 4-402 ReinstallServerConfig

Parameter	Mandatory	Type	Description
userTags	No	Array of UserTag objects	Cloud server labels. The key of a label must be unique. The maximum number of user-defined labels supported by CCE depends on the region. In the region that supports the least number of labels, you can still create up to 5 labels for a cloud server.
rootVolume	No	ReinstallVolumeSpec object	System disk configurations used in reinstallation.

Table 4-403 UserTag

Parameter	Mandatory	Type	Description
key	No	String	Key of the cloud server label. The value cannot start with CCE- or __type_baremetal .
value	No	String	Value of the cloud server label.

Table 4-404 ReinstallVolumeSpec

Parameter	Mandatory	Type	Description
imageID	No	String	Custom image ID.
cmkID	No	String	User master key ID. If this parameter is left blank by default, the EVS disk is not encrypted.

Table 4-405 ReinstallVolumeConfig

Parameter	Mandatory	Type	Description
lvmConfig	No	String	<p>Docker data disk configurations.</p> <p>The following is an example default configuration:</p> <pre>"lvmConfig": "dockerThinpool=vgpaas/90%VG;kubernetesLV=vgpaas/10%VG;diskType=evs;lvType=linear"</pre> <p>The following fields are included:</p> <ul style="list-style-type: none"> • userLV: size of the user space, for example, vgpaas/20%VG. • userPath: mount path of the user space, for example, /home/wqt-test. • diskType: disk type. Currently, only evs, hdd, and ssd are supported. • lvType: type of a logic volume. The value can be linear or striped. • dockerThinpool: Docker space size, for example, vgpaas/60%VG. • kubernetesLV: kubelet space size, for example, vgpaas/20%VG.

Parameter	Mandatory	Type	Description
storage	No	Storage object	<p>Disk initialization management parameter.</p> <p>This parameter is complex to configure. For details, see Attaching Disks to a Node.</p> <p>If this parameter retains its default, disks are managed based on the DockerLVMConfigOverride (discarded) parameter in extendParam. This parameter is supported by clusters of version 1.15.11 and later.</p> <p>NOTE</p> <ul style="list-style-type: none"> • If a node specification involves both local disks and EVS disks, do not retain the default value of this parameter to prevent unexpected disk partitions. • If you want to change the value range of a data disk to 20 to 32768, do not use the default value. • If you want to use the shared disk space (with the runtime and Kubernetes partitions cancelled), do not retain the default value of this parameter. For details about the shared disk space, see Data Disk Space Allocation.

Table 4-406 Storage

Parameter	Mandatory	Type	Description
storageSelectors	Yes	Array of StorageSelectors objects	Disk selection. Matched disks are managed according to matchLabels and storageType .
storageGroups	Yes	Array of StorageGroups objects	A storage group consists of multiple storage devices. It is used to divide storage space.

Table 4-407 StorageSelectors

Parameter	Mandatory	Type	Description
name	Yes	String	Selector name, used as the index of selectorNames in storageGroup . Therefore, the name of each selector must be unique.
storageType	Yes	String	Specifies the storage type. Currently, only evs (EVS volumes) and local (local volumes) are supported. The local storage does not support disk selection. All local disks will form a VG. Therefore, only one storageSelector of the local type is allowed.
matchLabels	No	matchLabels object	Matching field of an EVS volume. The size , volumeType , metadataEncrypted , metadataCmkid and count fields are supported.

Table 4-408 matchLabels

Parameter	Mandatory	Type	Description
size	No	String	Matched disk size. If this parameter is left unspecified, the disk size is not limited. Example: 100
volumeType	No	String	EVS disk type. SSD , GPSSD , SAS , ESSD , and SATA are supported. If this parameter is left blank, there is no restriction on the disk type.
metadataEncrypted	No	String	Disk encryption identifier. If this parameter is set to 0 , it specifies that the disk is not encrypted. If this parameter is set to 1 , it specifies that the disk is encrypted. If this parameter is left blank, there is no restriction on the disk encryption identifier.

Parameter	Mandatory	Type	Description
metadataCmkid	No	String	CMK ID of an encrypted disk. The value is a string of 36 bytes. If this parameter is left blank, there is no restriction on the disk key ID.
count	No	String	Number of disks to be selected. If this parameter is left blank, all disks of this type are selected.

Table 4-409 StorageGroups

Parameter	Mandatory	Type	Description
name	Yes	String	Name of a virtual storage group, which must be unique. NOTE <ul style="list-style-type: none"> If cceManaged is set to true, the name must be set to vgpaas. If the data disk is used as a temporary storage volume, the name must be vg-everest-localvolume-ephemeral. If the data disk is used as a persistent storage volume, the name must be vg-everest-localvolume-persistent.
cceManaged	No	Boolean	Storage space for Kubernetes and runtime components. Only one group can be set to true . If this parameter is left blank, the default value false is used.
selectorNames	Yes	Array of strings	This parameter corresponds to name in storageSelectors . A group can match multiple selectors, but a selector can match only one group.
virtualSpaces	Yes	Array of VirtualSpace objects	Detailed management of space configuration in a group.

Table 4-410 VirtualSpace

Parameter	Mandatory	Type	Description
name	Yes	String	Name of a virtualSpace. Options: share , kubernetes , runtime , and user <ul style="list-style-type: none"> • share: shared space configuration (with the runtime and Kubernetes partitions cancelled). lvmConfig needs to be configured. • Kubernetes: Kubernetes space configuration. lvmConfig needs to be configured. • runtime: runtime space configuration. runtimeConfig needs to be configured. • user: user space configuration. lvmConfig needs to be configured.
size	Yes	String	Size of a virtualSpace. The value must be an integer in percentage. Example: 90%. NOTE The sum of the percentages of all virtualSpaces in a group cannot exceed 100%.
lvmConfig	No	LVMConfig object	LVM configuration managements, applicable to the <i>share</i> *, kubernetes , and user spaces. Note that one virtual space supports only one config.
runtimeConfig	No	RuntimeConfig object	runtime configurations, applicable to the runtime space. Note that one virtual space supports only one config.

Table 4-411 LVMConfig

Parameter	Mandatory	Type	Description
lvType	Yes	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.
path	No	String	Path to which the disk is attached. This parameter takes effect only in user configuration. The value is an absolute path. Digits, letters, periods (.), hyphens (-), and underscores (_) are allowed.

Table 4-412 RuntimeConfig

Parameter	Mandatory	Type	Description
lvType	Yes	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.

Table 4-413 ReinstallRuntimeConfig

Parameter	Mandatory	Type	Description
dockerBaseSize	No	Integer	<p>Available disk space of a single container on a node, in GB. This parameter has been discarded. Use containerBaseSize.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In Device Mapper mode, the default value is 10. In OverlayFS mode, the available space of a single container is not limited by default, and the dockerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters of the new versions.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that dockerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device.</p>

Parameter	Mandatory	Type	Description
containerBaseSize	No	Integer	<p>Available disk space of a single container on a node, in GB.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In OverlayFS mode, the available space of a single container is not limited by default. In Device Mapper mode, the default value is 10, and the containerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters v1.23.14-r0, v1.25.9-r0, v1.27.6-r0, v1.28.4-r0 or later.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that containerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device. In new versions, the Device Mapper mode is used only by BMSs in the same resource pool and is being discarded.</p>

Parameter	Mandatory	Type	Description
runtime	No	Runtime object	Container runtime: <ul style="list-style-type: none"> • Clusters of v1.25 or earlier: docker. • Clusters of v1.25 or later: Container runtime varies with the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker. For nodes running other OSs, the default container runtime is containerd.

Table 4-414 Runtime

Parameter	Mandatory	Type	Description
name	No	String	Container runtime. Default value: <ul style="list-style-type: none"> • Clusters earlier than v1.25: The default value is docker. • Clusters of v1.25 or later: The default container runtime varies depending on the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker; for nodes running other OSs, the default container runtime is containerd.

Table 4-415 ReinstallK8sOptionsConfig

Parameter	Mandatory	Type	Description
labels	No	Map<String,String>	<p>Defined in key-value pairs. A maximum of 20 key-value pairs are allowed.</p> <ul style="list-style-type: none"> • Key: Enter 1 to 63 characters, starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain can be prefixed to a key and contain a maximum of 253 characters. Example DNS subdomain: example.com/my-key • Value: The value can be left blank or contain 1 to 63 characters that start with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed in the character string. <p>Example:</p> <pre>"k8sTags": { "key": "value" }</pre>

Parameter	Mandatory	Type	Description
taints	No	Array of Taint objects	<p>Taints can be added for anti-affinity when creating nodes. A maximum of 20 taints can be added. Each taint contains the following parameters:</p> <ul style="list-style-type: none"> • Key: A key must contain 1 to 63 characters starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain name can be used as the prefix of a key. • Value: A value must start with a letter or digit and can contain a maximum of 63 characters, including letters, digits, hyphens (-), underscores (_), and periods (.). • Effect: Available options are NoSchedule, PreferNoSchedule, and NoExecute. <p>Example:</p> <pre>"taints": [{ "key": "status", "value": "unavailable", "effect": "NoSchedule" }, { "key": "looks", "value": "bad", "effect": "NoSchedule" }]</pre>
maxPods	No	Integer	<p>Maximum number of pods that can be created on a node, including the default system pods. Value range: 16 to 256.</p> <p>This limit prevents the node from being overloaded of pods.</p>

Parameter	Mandatory	Type	Description
nicMultiqueue	No	String	<ul style="list-style-type: none"> Number of ENI queues. Example setting: <code>"queue":4</code> The following fields are included: queue: number of ENI queues. This field can be configured only for BMS nodes in a CCE Turbo cluster. Supported proportions are {"1":128, "2":92, "4":92, "8":32, "16":16, "28":9}. That is, if there is one queue, a maximum of 128 ENIs can be bound. If there are two queues, a maximum of 92 ENIs can be bound for two queues. A larger number of ENI queues indicates higher performance but fewer ENIs can be bound. The queue settings cannot be changed after the node pool is created.

Parameter	Mandatory	Type	Description
nicThreshold	No	String	<ul style="list-style-type: none"> ENI pre-binding thresholds. Example setting: "0.3:0.6" Low threshold (L): determines the minimum number of pre-bound ENIs (Min). Formula: $Min = Total\ number\ of\ ENIs\ of\ the\ node \times L$ High threshold (H): determines the maximum number of pre-bound ENIs (Max). Formula: $Max = Total\ number\ of\ ENIs\ of\ the\ node \times H$ Number of ENIs bound to a BMS node (B) and number of ENIs being used by pods (U): $U + Min < B < U + Max$ If the number of pre-bound ENIs on a BMS node is smaller than the minimum allowed, the system will bind more ENIs to make the numbers equal. If the number of pre-bound ENIs on a BMS node is larger than the maximum allowed, the system periodically unbinds ENIs (about every 2 minutes) to make the numbers equal. Both the thresholds are one-decimal-place values ranging from 0.0 to 1.0. The low threshold must be smaller than or equal to the high one. This field can be configured only for BMS nodes in a CCE Turbo cluster. Pre-binding ENIs can speed up workload creation but occupies IP addresses.

Table 4-416 Taint

Parameter	Mandatory	Type	Description
key	Yes	String	Key.
value	No	String	Value.
effect	Yes	String	Effect.

Table 4-417 NodeLifecycleConfig

Parameter	Mandatory	Type	Description
preInstall	No	String	Pre-installation script. NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)
postInstall	No	String	Post-installation script. NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)

Table 4-418 ReinstallExtendParam

Parameter	Mandatory	Type	Description
alpha.cce/ NodeImageID	No	String	(Discarded) ID of the user image to run the target OS. Specifying this parameter is equivalent to specifying imageID in ReinstallVolumeSpec . The original value will be overwritten.

Table 4-419 HostnameConfig

Parameter	Mandatory	Type	Description
type	Yes	String	<p>Configuration type of the Kubernetes node name. The default value is privatelp.</p> <ul style="list-style-type: none"> • privatelp: The Kubernetes node is named after its IP address. • cceNodeName: The Kubernetes node is named after the CCE node. <p>NOTE</p> <ul style="list-style-type: none"> • For a node which is configured using cceNodeName, the name is the same as the Kubernetes node name and the ECS name. The node name cannot be changed. If the ECS name is changed on the ECS console, the node name will retain unchanged after ECS synchronization. • For a node which is configured using cceNodeName, to avoid a conflict between Kubernetes nodes, the system automatically adds a suffix to each node name. The suffix is in the format of A hyphen (-) Five random characters. The value of the random characters is a lowercase letter or a digit ranging from 0 to 9.

Response Parameters

Status code: 200

Table 4-420 Response body parameters

Parameter	Type	Description
jobid	String	Job ID returned after the job is delivered. The job ID can be used to query the job execution status.

Example Requests

Add a node running EulerOS 2.5 to a cluster.

```
POST /api/v3/projects/{project_id}/clusters/{cluster_id}/nodes/add
{
  "kind": "List",
  "apiVersion": "v3",
  "nodeList": [ {
    "serverID": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "spec": {
      "name": "my-ecs-0001",
      "os": "EulerOS 2.5",
      "login": {
        "sshKey": "KeyPair-001"
      }
    }
  }
} ]
}
```

Example Responses

Status code: 200

The job for accepting a node into a specified cluster is successfully delivered.

```
{
  "jobid": "2ec9b78d-9368-46f3-8f29-d1a95622a568"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Add a node running EulerOS 2.5 to a cluster.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

import java.util.UUID;
import java.util.List;
import java.util.ArrayList;

public class AddNodeSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";
```

```
ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

CceClient client = CceClient.newBuilder()
    .withCredential(auth)
    .withRegion(CceRegion.valueOf("<YOUR REGION>"))
    .build();
AddNodeRequest request = new AddNodeRequest();
request.withClusterId("{cluster_id}");
AddNodeList body = new AddNodeList();
Login loginSpec = new Login();
loginSpec.withSshKey("KeyPair-001");
ReinstallNodeSpec specNodeList = new ReinstallNodeSpec();
specNodeList.withOs("EulerOS 2.5")
    .withLogin(loginSpec)
    .withName("my-ecs-0001");
List<AddNode> listbodyNodeList = new ArrayList<>();
listbodyNodeList.add(
    new AddNode()
        .withServerID(UUID.fromString("xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"))
        .withSpec(specNodeList)
);
body.withNodeList(listbodyNodeList);
body.withKind("List");
body.withApiVersion("v3");
request.withBody(body);
try {
    AddNodeResponse response = client.addNode(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

Add a node running EulerOS 2.5 to a cluster.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \
    client = CceClient.new_builder() \
```

```
.with_credentials(credentials) \  
.with_region(CceRegion.value_of("<YOUR REGION>")) \  
.build()  
  
try:  
    request = AddNodeRequest()  
    request.cluster_id = "{cluster_id}"  
    loginSpec = Login(  
        ssh_key="KeyPair-001"  
    )  
    specNodeList = ReinstallNodeSpec(  
        os="EulerOS 2.5",  
        login=loginSpec,  
        name="my-ecs-0001"  
    )  
    listNodeListbody = [  
        AddNode(  
            server_id="xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",  
            spec=specNodeList  
        )  
    ]  
    request.body = AddNodeList(  
        node_list=listNodeListbody,  
        kind="List",  
        api_version="v3"  
    )  
    response = client.add_node(request)  
    print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

Go

Add a node running EulerOS 2.5 to a cluster.

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()  
  
    client := cce.NewCceClient(  
        cce.CceClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())
```

```

request := &model.AddNodeRequest{}
request.ClusterId = "{cluster_id}"
sshKeyLogin:= "KeyPair-001"
loginSpec := &model.Login{
    SshKey: &sshKeyLogin,
}
nameSpec:= "my-ecs-0001"
specNodeList := &model.ReinstallNodeSpec{
    Os: "EulerOS 2.5",
    Login: loginSpec,
    Name: &nameSpec,
}
var listNodeListbody = []model.AddNode{
    {
        ServerID: "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
        Spec: specNodeList,
    },
}
request.Body = &model.AddNodeList{
    NodeList: listNodeListbody,
    Kind: "List",
    ApiVersion: "v3",
}
response, err := client.AddNode(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The job for accepting a node into a specified cluster is successfully delivered.

Error Codes

See [Error Codes](#).

4.3.7 Resetting a Node

Function

This API is used to reset a node in a specified cluster.

NOTE

The URL for cluster management is in the format of https://Endpoint/uri. In the URL, **uri** indicates the resource path, that is, the path for API access.

Calling Method

For details, see [Calling APIs](#).

URI

POST /api/v3/projects/{project_id}/clusters/{cluster_id}/nodes/reset

Table 4-421 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-422 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Table 4-423 Request body parameters

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	API version. The value is fixed at v3 .
kind	Yes	String	API type. The value is fixed at List .

Parameter	Mandatory	Type	Description
nodeList	Yes	Array of ResetNode objects	List of nodes to be reset. A maximum of 200 nodes can be reset at a time.

Table 4-424 ResetNode

Parameter	Mandatory	Type	Description
nodeID	Yes	String	Node ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
spec	Yes	ReinstallNodeSpec object	Node reinstallation configuration parameters. Nodes in a node pool cannot be specified externally. These nodes will be reinstalled based on the node pool settings. By default, this parameter is mandatory for nodes in a node pool.

Table 4-425 ReinstallNodeSpec

Parameter	Mandatory	Type	Description
os	Yes	String	Operating system. If you specify a custom image, the actual OS version in the IMS image is used. Select an OS version supported by the current cluster, for example, EulerOS 2.5, CentOS 7.6, or EulerOS 2.8.
login	Yes	Login object	Node login mode. Either the key pair or password must be used for login.

Parameter	Mandatory	Type	Description
name	No	String	Node name. NOTE Specifying this field during reinstallation will change the node name, and the server name will change accordingly. By default, the current server name is used as the node name. Enter 1 to 56 characters starting with a letter and not ending with a hyphen (-).
serverConfig	No	ReinstallServerConfig object	Server configuration.
volumeConfig	No	ReinstallVolumeConfig object	Volume management configuration.
runtimeConfig	No	ReinstallRuntimeConfig object	Container runtime configuration.
k8sOptions	No	ReinstallK8sOptionsConfig object	Kubernetes node configuration.
lifecycle	No	NodeLifecycleConfig object	Customized lifecycle configuration of a node.

Parameter	Mandatory	Type	Description
initializedConditions	No	Array of strings	<p>Custom initialization flag. Before CCE nodes are initialized, they are tainted with node.cloudprovider.kubernetes.io/uninitialized to prevent pods from being scheduled to them.</p> <p>CCE supports custom initialization flags. After receiving the initializedConditions parameter, CCE converts the parameter value into a node label and provisions the label with the node, for example, cloudprovider.openvessel.io/inject-initialized-conditions=CCEInitial_CustomedInitial.</p> <p>After the node is labeled, its status.Conditions is polled to check whether the type of conditions has a flag name, such as CCEInitial and CustomedInitial. If all input flags exist and their status is True, the node initialization is complete and the initialization taint is removed.</p> <ul style="list-style-type: none"> • Use only letters and digits. Max. characters: 20. • Max. flags: 2.
extendParam	No	ReinstallExtendParam object	Extended reinstallation parameter, which is discarded.
hostnameConfig	No	HostnameConfig object	Kubernetes node name configuration parameter, which is supported by clusters of v1.23.6-r0 to v1.25 or clusters of v1.25.2-r0 or later versions.

Table 4-426 Login

Parameter	Mandatory	Type	Description
sshKey	No	String	Name of the key pair used for login.
userPassword	No	UserPassword object	Password used for node login.

Table 4-427 UserPassword

Parameter	Mandatory	Type	Description
username	No	String	Login account. The default value is root .
password	Yes	String	<p>If a username and a password are used to create a node, this field is shielded in the response body.</p> <p>A password must meet the following complexity requirements:</p> <ul style="list-style-type: none"> • Contains 8 to 26 characters. • Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (!@\$%^-_=+[{ }];,./?~#*) • Cannot contain the username or the username spelled backwards. <p>The password field must be salted during node creation. For details, see Adding a Salt in the password Field When Creating a Node.</p>

Table 4-428 ReinstallServerConfig

Parameter	Mandatory	Type	Description
userTags	No	Array of UserTag objects	Cloud server labels. The key of a label must be unique. The maximum number of user-defined labels supported by CCE depends on the region. In the region that supports the least number of labels, you can still create up to 5 labels for a cloud server.
rootVolume	No	ReinstallVolumeSpec object	System disk configurations used in reinstallation.

Table 4-429 UserTag

Parameter	Mandatory	Type	Description
key	No	String	Key of the cloud server label. The value cannot start with CCE- or __type_baremetal .
value	No	String	Value of the cloud server label.

Table 4-430 ReinstallVolumeSpec

Parameter	Mandatory	Type	Description
imageID	No	String	Custom image ID.
cmkID	No	String	User master key ID. If this parameter is left blank by default, the EVS disk is not encrypted.

Table 4-431 ReinstallVolumeConfig

Parameter	Mandatory	Type	Description
lvmConfig	No	String	<p>Docker data disk configurations.</p> <p>The following is an example default configuration:</p> <pre>"lvmConfig": "dockerThinpool=vgpaas/90%VG;kubernetesLV=vgpaas/10%VG;diskType=evs;lvType=linear"</pre> <p>The following fields are included:</p> <ul style="list-style-type: none"> • userLV: size of the user space, for example, vgpaas/20%VG. • userPath: mount path of the user space, for example, /home/wqt-test. • diskType: disk type. Currently, only evs, hdd, and ssd are supported. • lvType: type of a logic volume. The value can be linear or striped. • dockerThinpool: Docker space size, for example, vgpaas/60%VG. • kubernetesLV: kubelet space size, for example, vgpaas/20%VG.

Parameter	Mandatory	Type	Description
storage	No	Storage object	<p>Disk initialization management parameter.</p> <p>This parameter is complex to configure. For details, see Attaching Disks to a Node.</p> <p>If this parameter retains its default, disks are managed based on the DockerLVMConfigOverride (discarded) parameter in extendParam. This parameter is supported by clusters of version 1.15.11 and later.</p> <p>NOTE</p> <ul style="list-style-type: none"> • If a node specification involves both local disks and EVS disks, do not retain the default value of this parameter to prevent unexpected disk partitions. • If you want to change the value range of a data disk to 20 to 32768, do not use the default value. • If you want to use the shared disk space (with the runtime and Kubernetes partitions cancelled), do not retain the default value of this parameter. For details about the shared disk space, see Data Disk Space Allocation.

Table 4-432 Storage

Parameter	Mandatory	Type	Description
storageSelectors	Yes	Array of StorageSelectors objects	Disk selection. Matched disks are managed according to matchLabels and storageType .
storageGroups	Yes	Array of StorageGroups objects	A storage group consists of multiple storage devices. It is used to divide storage space.

Table 4-433 StorageSelectors

Parameter	Mandatory	Type	Description
name	Yes	String	Selector name, used as the index of selectorNames in storageGroup . Therefore, the name of each selector must be unique.
storageType	Yes	String	Specifies the storage type. Currently, only evs (EVS volumes) and local (local volumes) are supported. The local storage does not support disk selection. All local disks will form a VG. Therefore, only one storageSelector of the local type is allowed.
matchLabels	No	matchLabels object	Matching field of an EVS volume. The size , volumeType , metadataEncrypted , metadataCmkid and count fields are supported.

Table 4-434 matchLabels

Parameter	Mandatory	Type	Description
size	No	String	Matched disk size. If this parameter is left unspecified, the disk size is not limited. Example: 100
volumeType	No	String	EVS disk type. SSD , GPSSD , SAS , ESSD , and SATA are supported. If this parameter is left blank, there is no restriction on the disk type.
metadataEncrypted	No	String	Disk encryption identifier. If this parameter is set to 0 , it specifies that the disk is not encrypted. If this parameter is set to 1 , it specifies that the disk is encrypted. If this parameter is left blank, there is no restriction on the disk encryption identifier.

Parameter	Mandatory	Type	Description
metadataCmkid	No	String	CMK ID of an encrypted disk. The value is a string of 36 bytes. If this parameter is left blank, there is no restriction on the disk key ID.
count	No	String	Number of disks to be selected. If this parameter is left blank, all disks of this type are selected.

Table 4-435 StorageGroups

Parameter	Mandatory	Type	Description
name	Yes	String	Name of a virtual storage group, which must be unique. NOTE <ul style="list-style-type: none"> If cceManaged is set to true, the name must be set to vgpaas. If the data disk is used as a temporary storage volume, the name must be vg-everest-localvolume-ephemeral. If the data disk is used as a persistent storage volume, the name must be vg-everest-localvolume-persistent.
cceManaged	No	Boolean	Storage space for Kubernetes and runtime components. Only one group can be set to true . If this parameter is left blank, the default value false is used.
selectorNames	Yes	Array of strings	This parameter corresponds to name in storageSelectors . A group can match multiple selectors, but a selector can match only one group.
virtualSpaces	Yes	Array of VirtualSpace objects	Detailed management of space configuration in a group.

Table 4-436 VirtualSpace

Parameter	Mandatory	Type	Description
name	Yes	String	Name of a virtualSpace. Options: share , kubernetes , runtime , and user <ul style="list-style-type: none"> • share: shared space configuration (with the runtime and Kubernetes partitions cancelled). lvmConfig needs to be configured. • Kubernetes: Kubernetes space configuration. lvmConfig needs to be configured. • runtime: runtime space configuration. runtimeConfig needs to be configured. • user: user space configuration. lvmConfig needs to be configured.
size	Yes	String	Size of a virtualSpace. The value must be an integer in percentage. Example: 90%. NOTE The sum of the percentages of all virtualSpaces in a group cannot exceed 100%.
lvmConfig	No	LVMConfig object	LVM configuration managements, applicable to the <i>share</i> *, kubernetes , and user spaces. Note that one virtual space supports only one config.
runtimeConfig	No	RuntimeConfig object	runtime configurations, applicable to the runtime space. Note that one virtual space supports only one config.

Table 4-437 LVMConfig

Parameter	Mandatory	Type	Description
lvType	Yes	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.
path	No	String	Path to which the disk is attached. This parameter takes effect only in user configuration. The value is an absolute path. Digits, letters, periods (.), hyphens (-), and underscores (_) are allowed.

Table 4-438 RuntimeConfig

Parameter	Mandatory	Type	Description
lvType	Yes	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.

Table 4-439 ReinstallRuntimeConfig

Parameter	Mandatory	Type	Description
dockerBaseSize	No	Integer	<p>Available disk space of a single container on a node, in GB. This parameter has been discarded. Use containerBaseSize.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In Device Mapper mode, the default value is 10. In OverlayFS mode, the available space of a single container is not limited by default, and the dockerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters of the new versions.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that dockerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device.</p>

Parameter	Mandatory	Type	Description
containerBaseSize	No	Integer	<p>Available disk space of a single container on a node, in GB.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In OverlayFS mode, the available space of a single container is not limited by default. In Device Mapper mode, the default value is 10, and the containerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters v1.23.14-r0, v1.25.9-r0, v1.27.6-r0, v1.28.4-r0 or later.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that containerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device. In new versions, the Device Mapper mode is used only by BMSs in the same resource pool and is being discarded.</p>

Parameter	Mandatory	Type	Description
runtime	No	Runtime object	Container runtime: <ul style="list-style-type: none"> • Clusters of v1.25 or earlier: docker. • Clusters of v1.25 or later: Container runtime varies with the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker. For nodes running other OSs, the default container runtime is containerd.

Table 4-440 Runtime

Parameter	Mandatory	Type	Description
name	No	String	Container runtime. Default value: <ul style="list-style-type: none"> • Clusters earlier than v1.25: The default value is docker. • Clusters of v1.25 or later: The default container runtime varies depending on the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker; for nodes running other OSs, the default container runtime is containerd.

Table 4-441 ReinstallK8sOptionsConfig

Parameter	Mandatory	Type	Description
labels	No	Map<String,String>	<p>Defined in key-value pairs. A maximum of 20 key-value pairs are allowed.</p> <ul style="list-style-type: none"> Key: Enter 1 to 63 characters, starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain can be prefixed to a key and contain a maximum of 253 characters. Example DNS subdomain: example.com/my-key Value: The value can be left blank or contain 1 to 63 characters that start with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed in the character string. <p>Example:</p> <pre>"k8sTags": { "key": "value" }</pre>

Parameter	Mandatory	Type	Description
taints	No	Array of Taint objects	<p>Taints can be added for anti-affinity when creating nodes. A maximum of 20 taints can be added. Each taint contains the following parameters:</p> <ul style="list-style-type: none"> • Key: A key must contain 1 to 63 characters starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain name can be used as the prefix of a key. • Value: A value must start with a letter or digit and can contain a maximum of 63 characters, including letters, digits, hyphens (-), underscores (_), and periods (.). • Effect: Available options are NoSchedule, PreferNoSchedule, and NoExecute. <p>Example:</p> <pre>"taints": [{ "key": "status", "value": "unavailable", "effect": "NoSchedule" }, { "key": "looks", "value": "bad", "effect": "NoSchedule" }]</pre>
maxPods	No	Integer	<p>Maximum number of pods that can be created on a node, including the default system pods. Value range: 16 to 256.</p> <p>This limit prevents the node from being overloaded of pods.</p>

Parameter	Mandatory	Type	Description
nicMultiqueue	No	String	<ul style="list-style-type: none"> Number of ENI queues. Example setting: <code>"queue":4</code> The following fields are included: queue: number of ENI queues. This field can be configured only for BMS nodes in a CCE Turbo cluster. Supported proportions are {"1":128, "2":92, "4":92, "8":32, "16":16, "28":9}. That is, if there is one queue, a maximum of 128 ENIs can be bound. If there are two queues, a maximum of 92 ENIs can be bound for two queues. A larger number of ENI queues indicates higher performance but fewer ENIs can be bound. The queue settings cannot be changed after the node pool is created.

Parameter	Mandatory	Type	Description
nicThreshold	No	String	<ul style="list-style-type: none"> ENI pre-binding thresholds. Example setting: "0.3:0.6" Low threshold (L): determines the minimum number of pre-bound ENIs (Min). Formula: $Min = Total\ number\ of\ ENIs\ of\ the\ node \times L$ High threshold (H): determines the maximum number of pre-bound ENIs (Max). Formula: $Max = Total\ number\ of\ ENIs\ of\ the\ node \times H$ Number of ENIs bound to a BMS node (B) and number of ENIs being used by pods (U): $U + Min < B < U + Max$ If the number of pre-bound ENIs on a BMS node is smaller than the minimum allowed, the system will bind more ENIs to make the numbers equal. If the number of pre-bound ENIs on a BMS node is larger than the maximum allowed, the system periodically unbinds ENIs (about every 2 minutes) to make the numbers equal. Both the thresholds are one-decimal-place values ranging from 0.0 to 1.0. The low threshold must be smaller than or equal to the high one. This field can be configured only for BMS nodes in a CCE Turbo cluster. Pre-binding ENIs can speed up workload creation but occupies IP addresses.

Table 4-442 Taint

Parameter	Mandatory	Type	Description
key	Yes	String	Key.
value	No	String	Value.
effect	Yes	String	Effect.

Table 4-443 NodeLifecycleConfig

Parameter	Mandatory	Type	Description
preInstall	No	String	Pre-installation script. NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)
postInstall	No	String	Post-installation script. NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)

Table 4-444 ReinstallExtendParam

Parameter	Mandatory	Type	Description
alpha.cce/ NodeImageID	No	String	(Discarded) ID of the user image to run the target OS. Specifying this parameter is equivalent to specifying imageID in ReinstallVolumeSpec . The original value will be overwritten.

Table 4-445 HostnameConfig

Parameter	Mandatory	Type	Description
type	Yes	String	<p>Configuration type of the Kubernetes node name. The default value is privatelp.</p> <ul style="list-style-type: none"> • privatelp: The Kubernetes node is named after its IP address. • cceNodeName: The Kubernetes node is named after the CCE node. <p>NOTE</p> <ul style="list-style-type: none"> • For a node which is configured using cceNodeName, the name is the same as the Kubernetes node name and the ECS name. The node name cannot be changed. If the ECS name is changed on the ECS console, the node name will retain unchanged after ECS synchronization. • For a node which is configured using cceNodeName, to avoid a conflict between Kubernetes nodes, the system automatically adds a suffix to each node name. The suffix is in the format of A hyphen (-) Five random characters. The value of the random characters is a lowercase letter or a digit ranging from 0 to 9.

Response Parameters

Status code: 200

Table 4-446 Response body parameters

Parameter	Type	Description
jobid	String	Job ID returned after the job is delivered. The job ID can be used to query the job execution status.

Example Requests

- Reset the nodes running EulerOS 2.5 in the default node pool.

POST /api/v3/projects/{project_id}/clusters/{cluster_id}/nodes/reset

```
{
  "kind": "List",
  "apiVersion": "v3",
  "nodeList": [ {
    "nodeID": "yyyyyyyy-yyy-yyy-yyy-yyyyyyyyyyyy",
    "spec": {
      "name": "my-ecs-0001",
      "os": "EulerOS 2.5",
      "login": {
        "sshKey": "KeyPair-001"
      }
    }
  }
  ]
}
```

- Reset a node in a node pool (invalid spec).

POST /api/v3/projects/{project_id}/clusters/{cluster_id}/nodes/reset

```
{
  "kind": "List",
  "apiVersion": "v3",
  "nodeList": [ {
    "nodeID": "xxxxxxxx-xxx-xxx-xxx-xxxxxxxxxxxx",
    "spec": {
      "name": "my-ecs-0001",
      "os": "EulerOS 2.5",
      "login": {
        "sshKey": "KeyPair-001"
      }
    }
  }
  ]
}
```

Example Responses

Status code: 200

The job for resetting a node in a specified cluster is successfully delivered.

```
{
  "jobid": "2ec9b78d-9368-46f3-8f29-d1a95622a568"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

- Reset the nodes running EulerOS 2.5 in the default node pool.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;
```

```
import java.util.UUID;
import java.util.List;
import java.util.ArrayList;

public class ResetNodeSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();

        ResetNodeRequest request = new ResetNodeRequest();
        request.withClusterId("{cluster_id}");
        ResetNodeList body = new ResetNodeList();
        Login loginSpec = new Login();
        loginSpec.withSshKey("KeyPair-001");
        ReinstallNodeSpec specNodeList = new ReinstallNodeSpec();
        specNodeList.withOs("EulerOS 2.5")
            .withLogin(loginSpec)
            .withName("my-ecs-0001");
        List<ResetNode> listbodyNodeList = new ArrayList<>();
        listbodyNodeList.add(
            new ResetNode()
                .withNodeID(UUID.fromString("yyyyyyyy-yy-yy-yy-yyy-yyyy-yyyy-yyyy"))
                .withSpec(specNodeList)
        );
        body.withNodeList(listbodyNodeList);
        body.withKind("List");
        body.withApiVersion("v3");
        request.withBody(body);
        try {
            ResetNodeResponse response = client.resetNode(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- **Reset a node in a node pool (invalid spec).**

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
```

```
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

import java.util.UUID;
import java.util.List;
import java.util.ArrayList;

public class ResetNodeSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        ResetNodeRequest request = new ResetNodeRequest();
        request.withClusterId("{cluster_id}");
        ResetNodeList body = new ResetNodeList();
        Login loginSpec = new Login();
        loginSpec.withSshKey("KeyPair-001");
        ReinstallNodeSpec specNodeList = new ReinstallNodeSpec();
        specNodeList.withOs("EulerOS 2.5")
            .withLogin(loginSpec)
            .withName("my-ecs-0001");
        List<ResetNode> listbodyNodeList = new ArrayList<>();
        listbodyNodeList.add(
            new ResetNode()
                .withNodeID(UUID.fromString("xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"))
                .withSpec(specNodeList)
        );
        body.withNodeList(listbodyNodeList);
        body.withKind("List");
        body.withApiVersion("v3");
        request.withBody(body);
        try {
            ResetNodeResponse response = client.resetNode(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- Reset the nodes running EulerOS 2.5 in the default node pool.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ResetNodeRequest()
        request.cluster_id = "{cluster_id}"
        loginSpec = Login(
            ssh_key="KeyPair-001"
        )
        specNodeList = ReinstallNodeSpec(
            os="EulerOS 2.5",
            login=loginSpec,
            name="my-ecs-0001"
        )
        listNodeListbody = [
            ResetNode(
                node_id="yyyyyyyy-yy-yy-yy-yy-yy-yy-yy-yy",
                spec=specNodeList
            )
        ]
        request.body = ResetNodeList(
            node_list=listNodeListbody,
            kind="List",
            api_version="v3"
        )
        response = client.reset_node(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- **Reset a node in a node pool (invalid spec).**

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
```

```
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId) \

client = CceClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(CceRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ResetNodeRequest()
    request.cluster_id = "{cluster_id}"
    loginSpec = Login(
        ssh_key="KeyPair-001"
    )
    specNodeList = ReinstallNodeSpec(
        os="EulerOS 2.5",
        login=loginSpec,
        name="my-ecs-0001"
    )
    listNodeListbody = [
        ResetNode(
            node_id="xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
            spec=specNodeList
        )
    ]
    request.body = ResetNodeList(
        node_list=listNodeListbody,
        kind="List",
        api_version="v3"
    )
    response = client.reset_node(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

- Reset the nodes running EulerOS 2.5 in the default node pool.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
```

```

Build()

client := cce.NewCceClient(
    cce.CceClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ResetNodeRequest{
    request.ClusterId = "{cluster_id}"
    sshKeyLogin:= "KeyPair-001"
    loginSpec := &model.Login{
        SshKey: &sshKeyLogin,
    }
    nameSpec:= "my-ecs-0001"
    specNodeList := &model.ReinstallNodeSpec{
        Os: "EulerOS 2.5",
        Login: loginSpec,
        Name: &nameSpec,
    }
    var listNodeListbody = []model.ResetNode{
        {
            NodeId: "yyyyyyyy-yyy-yyy-yyyy-yyyyyyyyyyyy",
            Spec: specNodeList,
        },
    }
    request.Body = &model.ResetNodeList{
        NodeList: listNodeListbody,
        Kind: "List",
        ApiVersion: "v3",
    }
    response, err := client.ResetNode(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

- Reset a node in a node pool (invalid spec).

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).

```



```

        WithCredential(auth).
        Build()

        request := &model.ResetNodeRequest{
        request.ClusterId = "{cluster_id}"
        sshKeyLogin:= "KeyPair-001"
        loginSpec := &model.Login{
            SshKey: &sshKeyLogin,
        }
        nameSpec:= "my-ecs-0001"
        specNodeList := &model.ReinstallNodeSpec{
            Os: "EulerOS 2.5",
            Login: loginSpec,
            Name: &nameSpec,
        }
        var listNodeListbody = []model.ResetNode{
            {
                NodeID: "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
                Spec: specNodeList,
            },
        }
        request.Body = &model.ResetNodeList{
            NodeList: listNodeListbody,
            Kind: "List",
            ApiVersion: "v3",
        }
        response, err := client.ResetNode(request)
        if err == nil {
            fmt.Printf("%+v\n", response)
        } else {
            fmt.Println(err)
        }
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The job for resetting a node in a specified cluster is successfully delivered.

Error Codes

See [Error Codes](#).

4.3.8 Removing a Node

Function

This API is used to remove a node from a specified cluster.

 NOTE

The URL for cluster management is in the format of https://Endpoint/uri. In the URL, **uri** indicates the resource path, that is, the path for API access.

Calling Method

For details, see [Calling APIs](#).

URI

PUT /api/v3/projects/{project_id}/clusters/{cluster_id}/nodes/operation/remove

Table 4-447 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-448 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Table 4-449 Request body parameters

Parameter	Mandatory	Type	Description
apiVersion	No	String	API version. The value is fixed at v3 .

Parameter	Mandatory	Type	Description
kind	No	String	API type. The value is fixed at RemoveNodesTask .
spec	Yes	RemoveNodeSpec object	Configuration information.
status	No	TaskStatus object	Job status.

Table 4-450 RemoveNodesSpec

Parameter	Mandatory	Type	Description
login	Yes	Login object	Node login mode. Either the key pair or password must be used for login.
nodes	Yes	Array of NodeItem objects	List of nodes to be removed. A maximum of 200 nodes can be removed at a time.

Table 4-451 Login

Parameter	Mandatory	Type	Description
sshKey	No	String	Name of the key pair used for login.
userPassword	No	UserPassword object	Password used for node login.

Table 4-452 UserPassword

Parameter	Mandatory	Type	Description
username	No	String	Login account. The default value is root .

Parameter	Mandatory	Type	Description
password	Yes	String	<p>If a username and a password are used to create a node, this field is shielded in the response body.</p> <p>A password must meet the following complexity requirements:</p> <ul style="list-style-type: none"> • Contains 8 to 26 characters. • Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (!@\$%^_-=+[{ }],./?~#*) • Cannot contain the username or the username spelled backwards. <p>The password field must be salted during node creation. For details, see Adding a Salt in the password Field When Creating a Node.</p>

Table 4-453 Nodeltem

Parameter	Mandatory	Type	Description
uid	Yes	String	Node ID.

Table 4-454 TaskStatus

Parameter	Mandatory	Type	Description
jobID	No	String	Job ID, which is used by the caller to query the job progress.

Response Parameters

Status code: 200

Table 4-455 Response body parameters

Parameter	Type	Description
apiVersion	String	API version. The value is fixed at v3 .
kind	String	API type. The value is fixed at RemoveNodesTask .
spec	RemoveNodeSpec object	Configuration information.
status	TaskStatus object	Job status.

Table 4-456 RemoveNodesSpec

Parameter	Type	Description
login	Login object	Node login mode. Either the key pair or password must be used for login.
nodes	Array of NodeItem objects	List of nodes to be removed. A maximum of 200 nodes can be removed at a time.

Table 4-457 Login

Parameter	Type	Description
sshKey	String	Name of the key pair used for login.
userPassword	UserPassword object	Password used for node login.

Table 4-458 UserPassword

Parameter	Type	Description
username	String	Login account. The default value is root .

Parameter	Type	Description
password	String	<p>If a username and a password are used to create a node, this field is shielded in the response body.</p> <p>A password must meet the following complexity requirements:</p> <ul style="list-style-type: none"> • Contains 8 to 26 characters. • Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (!@\$%^-_=+[]{};,:./?~#*) • Cannot contain the username or the username spelled backwards. <p>The password field must be salted during node creation. For details, see Adding a Salt in the password Field When Creating a Node.</p>

Table 4-459 Nodeltem

Parameter	Type	Description
uid	String	Node ID.

Table 4-460 TaskStatus

Parameter	Type	Description
jobID	String	Job ID, which is used by the caller to query the job progress.

Example Requests

Removing a Node

```
PUT /api/v3/projects/{project_id}/clusters/{cluster_id}/nodes/operation/remove
```

```
{
  "spec": {
    "login": {
      "sshKey": "KeyPair-001"
    },
    "nodes": [ {
      "uid": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
    }, {
      "uid": "yyyyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyyy"
    } ]
  }
}
```

Example Responses

Status code: 200

The job for removing a node in a specified cluster is successfully delivered.

```
{
  "spec" : {
    "login" : {
      "sshKey" : "KeyPair-001"
    },
    "nodes" : [ {
      "uid" : "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
    }, {
      "uid" : "yyyyyyyy-yyyy-yyyy-yyyyyyyyyyyy"
    } ]
  },
  "status" : {
    "jobID" : "2ec9b78d-9368-46f3-8f29-d1a95622a568"
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Removing a Node

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

import java.util.List;
import java.util.ArrayList;

public class RemoveNodeSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        RemoveNodeRequest request = new RemoveNodeRequest();
        request.withClusterId("{cluster_id}");
```

```
RemoveNodesTask body = new RemoveNodesTask();
List<NodeItem> listSpecNodes = new ArrayList<>();
listSpecNodes.add(
    new NodeItem()
        .withUid("xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx")
);
listSpecNodes.add(
    new NodeItem()
        .withUid("yyyyyyyy-yyyy-yyyy-yyyy-yyy-yyyyyyyy")
);
Login loginSpec = new Login();
loginSpec.withSshKey("KeyPair-001");
RemoveNodesSpec specbody = new RemoveNodesSpec();
specbody.withLogin(loginSpec)
        .withNodes(listSpecNodes);
body.withSpec(specbody);
request.withBody(body);
try {
    RemoveNodeResponse response = client.removeNode(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

Removing a Node

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *
```

```
if __name__ == "__main__":
```

```
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
```

```
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"
```

```
    credentials = BasicCredentials(ak, sk, projectId) \
```

```
        client = CceClient.new_builder() \
            .with_credentials(credentials) \
            .with_region(CceRegion.value_of("<YOUR REGION>")) \
            .build()
```

```
    try:
```

```
        request = RemoveNodeRequest()
        request.cluster_id = "{cluster_id}"
        listNodesSpec = [
            NodeItem(
                uid="xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
            ),
```



```
        NodeItem(  
            uid="yyyyyyyy-yyy-yyy-yyy-yyy-yyyyyyyyyyyy"  
        )  
    ]  
    loginSpec = Login(  
        ssh_key="KeyPair-001"  
    )  
    specbody = RemoveNodesSpec(  
        login=loginSpec,  
        nodes=listNodesSpec  
    )  
    request.body = RemoveNodesTask(  
        spec=specbody  
    )  
    response = client.remove_node(request)  
    print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

Go

Removing a Node

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()  
  
    client := cce.NewCceClient(  
        cce.CceClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build()  
    )  
  
    request := &model.RemoveNodeRequest{}  
    request.ClusterId = "{cluster_id}"  
    var listNodesSpec = []model.NodeItem{  
        {  
            Uid: "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",  
        },  
        {  
            Uid: "yyyyyyyy-yyy-yyy-yyy-yyy-yyyyyyyyyyyy",  
        },  
    }  
    sshKeyLogin:= "KeyPair-001"
```

```
loginSpec := &model.Login{
    SshKey: &sshKeyLogin,
}
specbody := &model.RemoveNodesSpec{
    Login: loginSpec,
    Nodes: listNodesSpec,
}
request.Body = &model.RemoveNodesTask{
    Spec: specbody,
}
response, err := client.RemoveNode(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The job for removing a node in a specified cluster is successfully delivered.

Error Codes

See [Error Codes](#).

4.3.9 Migrating a Node

Function

This API is used to migrate a node from a specified cluster to another cluster. The node can be migrated only between different clusters in the same VPC and project, and the cluster types before and after the migration must be the same.

BMS nodes in a CCE Turbo cluster cannot be migrated.

NOTE

The URL for cluster management is in the format of **https://Endpoint/uri**, where **uri** indicates the resource path for API access.

Constraints

- Data can be migrated only between clusters in the same VPC and project.
- Data cannot be migrated between CCE Turbo clusters and CCE standard clusters.

- Data cannot be migrated between DEC and non-DEC clusters.
- Data cannot be migrated between clusters with IPv6 enabled and clusters without IPv6 enabled.
- Data can be migrated only to the default node pool of the target cluster.
- Physical ECS nodes in a CCE Turbo cluster cannot be migrated.

Calling Method

For details, see [Calling APIs](#).

URI

PUT /api/v3/projects/{project_id}/clusters/{cluster_id}/nodes/operation/migrateto/{target_cluster_id}

Table 4-461 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
target_cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-462 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Table 4-463 Request body parameters

Parameter	Mandatory	Type	Description
apiVersion	No	String	API version. The value is fixed at v3 .
kind	No	String	API type. The value is fixed at MigrateNodesTask .
spec	Yes	MigrateNodeSpec object	Configuration data.
status	No	TaskStatus object	Job status.

Table 4-464 MigrateNodesSpec

Parameter	Mandatory	Type	Description
os	Yes	String	Operating system type, which must be accurate to the version number. When alpha.cce/NodeImageID is specified, the value of os must be the same as the OS of the custom image.
extendParam	No	MigrateNodeExtendParam object	Extended parameters for node migration.
login	Yes	Login object	Node login mode. Either the key pair or password must be used for login.
runtime	No	Runtime object	Container runtime
nodes	Yes	Array of NodeItem objects	List of nodes to be migrated. A maximum of 200 nodes can be migrated at a time.

Table 4-465 MigrateNodeExtendParam

Parameter	Mandatory	Type	Description
maxPods	No	Integer	Maximum number of pods that can be created on a node, including the default system pods. Value range: 16 to 256. This limit prevents the node from being overloaded of pods.

Parameter	Mandatory	Type	Description
DockerLVMConfigOverride	No	String	<p>Docker data disk configurations.</p> <p>The disk type of the nodes to be migrated must be the same as that specified during node creation (that is, the value of diskType in DockerLVMConfigOverride must be the same as that specified during node creation). Ensure that the disk types of the nodes selected for a single API call are the same.</p> <p>The following is the default configuration:</p> <pre>"DockerLVMConfigOverride": "dockerThinpool=vgpaas/90%VG;kubernetesLV=vgpaas/10%VG;diskType=evs;lvType=linear"</pre> <p>The configuration contains the following fields:</p> <ul style="list-style-type: none"> • userLV (optional): size of the user space, for example, vgpaas/20%VG. • (Optional) userPath: mount path of the user space, for example, /home/wqt-test. • diskType: specifies the disk type. Currently, only evs, hdd, and ssd are supported. • lvType: type of the logical volume. Currently, the value can be linear or striped. For example, striped. • dockerThinpool: Docker disk size, for example, vgpaas/60%VG. • kubernetesLV: Kubelet space size, for example, vgpaas/20%VG.
alpha.cce/preInstall	No	String	<p>Pre-installation script.</p> <p>NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)</p>

Parameter	Mandatory	Type	Description
alpha.cce/postInstall	No	String	Post-installation script. NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)
alpha.cce/NodeImageID	No	String	ID of the user image to run the target OS. When alpha.cce/NodeImageID is specified, the value of os must be the same as the OS of the custom image.

Table 4-466 Login

Parameter	Mandatory	Type	Description
sshKey	No	String	Name of the key pair used for login.
userPassword	No	UserPassword object	Password used for node login.

Table 4-467 UserPassword

Parameter	Mandatory	Type	Description
username	No	String	Login account. The default value is root .

Parameter	Mandatory	Type	Description
password	Yes	String	<p>If a username and a password are used to create a node, this field is shielded in the response body.</p> <p>A password must meet the following complexity requirements:</p> <ul style="list-style-type: none"> • Contains 8 to 26 characters. • Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (!@\$%^-_=+[{ }];,./?~#*) • Cannot contain the username or the username spelled backwards. <p>The password field must be salted during node creation. For details, see Adding a Salt in the password Field When Creating a Node.</p>

Table 4-468 Runtime

Parameter	Mandatory	Type	Description
name	No	String	<p>Container runtime. Default value:</p> <ul style="list-style-type: none"> • Clusters earlier than v1.25: The default value is docker. • Clusters of v1.25 or later: The default container runtime varies depending on the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker; for nodes running other OSs, the default container runtime is containerd.

Table 4-469 NodeItem

Parameter	Mandatory	Type	Description
uid	Yes	String	Node ID.

Table 4-470 TaskStatus

Parameter	Mandatory	Type	Description
jobID	No	String	Job ID, which is used by the caller to query the job progress.

Response Parameters

Status code: 200

Table 4-471 Response body parameters

Parameter	Type	Description
apiVersion	String	API version. The value is fixed at v3 .
kind	String	API type. The value is fixed at MigrateNodesTask .
spec	MigrateNodeSpec object	Configuration data.
status	TaskStatus object	Job status.

Table 4-472 MigrateNodesSpec

Parameter	Type	Description
os	String	Operating system type, which must be accurate to the version number. When alpha.cce/NodeImageID is specified, the value of os must be the same as the OS of the custom image.
extendParam	MigrateNodeExtendParam object	Extended parameters for node migration.
login	Login object	Node login mode. Either the key pair or password must be used for login.

Parameter	Type	Description
runtime	Runtime object	Container runtime
nodes	Array of NodeItem objects	List of nodes to be migrated. A maximum of 200 nodes can be migrated at a time.

Table 4-473 MigrateNodeExtendParam

Parameter	Type	Description
maxPods	Integer	<p>Maximum number of pods that can be created on a node, including the default system pods. Value range: 16 to 256.</p> <p>This limit prevents the node from being overloaded of pods.</p>
DockerLVMConfigOverride	String	<p>Docker data disk configurations.</p> <p>The disk type of the nodes to be migrated must be the same as that specified during node creation (that is, the value of diskType in DockerLVMConfigOverride must be the same as that specified during node creation). Ensure that the disk types of the nodes selected for a single API call are the same.</p> <p>The following is the default configuration: <code>"DockerLVMConfigOverride": "dockerThinpool=vgpaas/90%VG;kubernetesLV=vgpaas/10%VG;diskType=evs;lvType=linear"</code></p> <p>The configuration contains the following fields:</p> <ul style="list-style-type: none"> • userLV (optional): size of the user space, for example, vgpaas/20%VG. • (Optional) userPath: mount path of the user space, for example, /home/wqt-test. • diskType: specifies the disk type. Currently, only evs, hdd, and ssd are supported. • lvType: type of the logical volume. Currently, the value can be linear or striped. For example, striped. • dockerThinpool: Docker disk size, for example, vgpaas/60%VG. • kubernetesLV: Kubelet space size, for example, vgpaas/20%VG.

Parameter	Type	Description
alpha.cce/ preInstall	String	Pre-installation script. NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)
alpha.cce/ postInstall	String	Post-installation script. NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)
alpha.cce/ NodeImageID	String	ID of the user image to run the target OS. When alpha.cce/NodeImageID is specified, the value of os must be the same as the OS of the custom image.

Table 4-474 Login

Parameter	Type	Description
sshKey	String	Name of the key pair used for login.
userPassword	UserPassword d object	Password used for node login.

Table 4-475 UserPassword

Parameter	Type	Description
username	String	Login account. The default value is root .

Parameter	Type	Description
password	String	<p>If a username and a password are used to create a node, this field is shielded in the response body.</p> <p>A password must meet the following complexity requirements:</p> <ul style="list-style-type: none"> • Contains 8 to 26 characters. • Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (!@\$%^-_=+[{ } ; , / ? ~ # *) • Cannot contain the username or the username spelled backwards. <p>The password field must be salted during node creation. For details, see Adding a Salt in the password Field When Creating a Node.</p>

Table 4-476 Runtime

Parameter	Type	Description
name	String	<p>Container runtime. Default value:</p> <ul style="list-style-type: none"> • Clusters earlier than v1.25: The default value is docker. • Clusters of v1.25 or later: The default container runtime varies depending on the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker; for nodes running other OSs, the default container runtime is containerd.

Table 4-477 Nodeltem

Parameter	Type	Description
uid	String	Node ID.

Table 4-478 TaskStatus

Parameter	Type	Description
jobID	String	Job ID, which is used by the caller to query the job progress.

Example Requests

- Migrate a node running EulerOS 2.5 to another cluster.

```
PUT /api/v3/projects/{project_id}/clusters/{cluster_id}/nodes/operation/migrateto/{target_cluster_id}
```

```
{
  "spec" : {
    "os" : "EulerOS 2.5",
    "login" : {
      "sshKey" : "KeyPair-001"
    },
    "nodes" : [ {
      "uid" : "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
    }, {
      "uid" : "yyyyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyyy"
    } ]
  }
}
```

- Migrate a node running EulerOS 2.5 and created using a specified image ID to another cluster.

```
PUT /api/v3/projects/{project_id}/clusters/{cluster_id}/nodes/operation/migrateto/{target_cluster_id}
```

```
{
  "spec" : {
    "os" : "EulerOS 2.5",
    "extendParam" : {
      "alpha.cce/NodeImageID" : "cc697ad7-9563-11e8-8ea7-0255ac106311"
    },
    "login" : {
      "sshKey" : "KeyPair-001"
    },
    "nodes" : [ {
      "uid" : "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
    }, {
      "uid" : "yyyyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyyy"
    } ]
  }
}
```

Example Responses

Status code: 200

The job for migrating a node from a specified cluster to another cluster is successfully delivered.

```
{
  "spec" : {
    "os" : "EulerOS 2.5",
    "login" : {
      "sshKey" : "KeyPair-001"
    },
    "runtime" : {
      "name" : "docker"
    },
    "nodes" : [ {
      "uid" : "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
    }, {
      "uid" : "yyyyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyyy"
    } ]
  },
  "status" : {
    "jobID" : "2ec9b78d-9368-46f3-8f29-d1a95622a568"
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

- Migrate a node running EulerOS 2.5 to another cluster.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

import java.util.List;
import java.util.ArrayList;

public class MigrateNodeSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        MigrateNodeRequest request = new MigrateNodeRequest();
        request.withClusterId("{cluster_id}");
        request.withTargetClusterId("{target_cluster_id}");
        MigrateNodesTask body = new MigrateNodesTask();
        List<NodeItem> listSpecNodes = new ArrayList<>();
        listSpecNodes.add(
            new NodeItem()
                .withUid("xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx")
        );
        listSpecNodes.add(
            new NodeItem()
                .withUid("yyyyyyyy-yy-yy-yy-yyy-yyyyyyyyyyyy")
        );
        Login loginSpec = new Login();
        loginSpec.withSshKey("KeyPair-001");
        MigrateNodesSpec specbody = new MigrateNodesSpec();
        specbody.withOs("EulerOS 2.5")
            .withLogin(loginSpec)
            .withNodes(listSpecNodes);
        body.withSpec(specbody);
        request.withBody(body);
        try {
            MigrateNodeResponse response = client.migrateNode(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
```

```
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

- Migrate a node running EulerOS 2.5 and created using a specified image ID to another cluster.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

import java.util.List;
import java.util.ArrayList;

public class MigrateNodeSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        MigrateNodeRequest request = new MigrateNodeRequest();
        request.withClusterId("{cluster_id}");
        request.withTargetClusterId("{target_cluster_id}");
        MigrateNodesTask body = new MigrateNodesTask();
        List<NodeItem> listSpecNodes = new ArrayList<>();
        listSpecNodes.add(
            new NodeItem()
                .withUid("xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx")
        );
        listSpecNodes.add(
            new NodeItem()
                .withUid("yyyyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyyy")
        );
        Login loginSpec = new Login();
        loginSpec.withSshKey("KeyPair-001");
        MigrateNodeExtendParam extendParamSpec = new MigrateNodeExtendParam();
        extendParamSpec.withAlphaCceNodeImageID("cc697ad7-9563-11e8-8ea7-0255ac106311");
        MigrateNodesSpec specbody = new MigrateNodesSpec();
```

```
specbody.withOs("EulerOS 2.5")
    .withExtendParam(extendParamSpec)
    .withLogin(loginSpec)
    .withNodes(listSpecNodes);
body.withSpec(specbody);
request.withBody(body);
try {
    MigrateNodeResponse response = client.migrateNode(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

- Migrate a node running EulerOS 2.5 to another cluster.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = MigrateNodeRequest()
        request.cluster_id = "{cluster_id}"
        request.target_cluster_id = "{target_cluster_id}"
        listNodesSpec = [
            NodelItem(
                uid="xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
            ),
            NodelItem(
                uid="yyyyyyyy-yy-yy-yy-yyy-yyyyyyyyyy"
            )
        ]
        loginSpec = Login(
            ssh_key="KeyPair-001"
        )
        specbody = MigrateNodesSpec(
            os="EulerOS 2.5",
            login=loginSpec,
```



```

        nodes=listNodesSpec
    )
    request.body = MigrateNodesTask(
        spec=specbody
    )
    response = client.migrate_node(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)

```

- Migrate a node running EulerOS 2.5 and created using a specified image ID to another cluster.

coding: utf-8

```

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = MigrateNodeRequest()
        request.cluster_id = "{cluster_id}"
        request.target_cluster_id = "{target_cluster_id}"
        listNodesSpec = [
            NodeItem(
                uid="xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
            ),
            NodeItem(
                uid="yyyyyyyy-yyy-yyy-yyy-yyyyyyyyyyyy"
            )
        ]
        loginSpec = Login(
            ssh_key="KeyPair-001"
        )
        extendParamSpec = MigrateNodeExtendParam(
            alpha_cce_node_image_id="cc697ad7-9563-11e8-8ea7-0255ac106311"
        )
        specbody = MigrateNodesSpec(
            os="EulerOS 2.5",
            extend_param=extendParamSpec,
            login=loginSpec,
            nodes=listNodesSpec
        )
        request.body = MigrateNodesTask(
            spec=specbody
        )
        response = client.migrate_node(request)
        print(response)

```

```
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

- Migrate a node running EulerOS 2.5 to another cluster.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.MigrateNodeRequest{}
    request.ClusterId = "{cluster_id}"
    request.TargetClusterId = "{target_cluster_id}"
    var listNodesSpec = []model.NodeItem{
        {
            Uid: "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
        },
        {
            Uid: "yyyyyyyy-yyy-yyy-yyy-yyyyyyyyyyyy",
        },
    }
    sshKeyLogin := "KeyPair-001"
    loginSpec := &model.Login{
        SshKey: &sshKeyLogin,
    }
    specbody := &model.MigrateNodesSpec{
        Os: "EulerOS 2.5",
        Login: loginSpec,
        Nodes: listNodesSpec,
    }
    request.Body = &model.MigrateNodesTask{
        Spec: specbody,
    }
    response, err := client.MigrateNode(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
```

```

    fmt.Println(err)
  }
}

```

- Migrate a node running EulerOS 2.5 and created using a specified image ID to another cluster.

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.MigrateNodeRequest{}
    request.ClusterId = "{cluster_id}"
    request.TargetClusterId = "{target_cluster_id}"
    var listNodesSpec = []model.NodeItem{
        {
            Uid: "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
        },
        {
            Uid: "yyyyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyyy",
        },
    }
    sshKeyLogin := "KeyPair-001"
    loginSpec := &model.Login{
        SshKey: &sshKeyLogin,
    }
    alphaCceNodeImageIdExtendParam := "cc697ad7-9563-11e8-8ea7-0255ac106311"
    extendParamSpec := &model.MigrateNodeExtendParam{
        AlphaCceNodeImageId: &alphaCceNodeImageIdExtendParam,
    }
    specbody := &model.MigrateNodesSpec{
        Os: "EulerOS 2.5",
        ExtendParam: extendParamSpec,
        Login: loginSpec,
        Nodes: listNodesSpec,
    }
    request.Body = &model.MigrateNodesTask{
        Spec: specbody,
    }
    response, err := client.MigrateNode(request)
    if err == nil {

```

```

    fmt.Printf("%+v\n", response)
  } else {
    fmt.Println(err)
  }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The job for migrating a node from a specified cluster to another cluster is successfully delivered.

Error Codes

See [Error Codes](#).

4.4 Node Pool Management

4.4.1 Creating a Node Pool

Function

This API is used to create a node pool in a specified cluster. This API can be called only when the cluster is in the available, scaling-out, or scaling-in state.

When creating a node pool in a cluster of v1.21, you can bind security groups to the node pool. A maximum of five security groups can be bound to a node pool.

After the security groups of a node pool are updated, the update takes effect only for newly created pods. You are advised to evict the original pods on the node.

NOTE

If there is no cluster, [create one](#).

The URL for cluster management is in the format of **https://Endpoint/uri**. In the URL, **uri** indicates the resource path, that is, the path for API access.

Calling Method

For details, see [Calling APIs](#).

URI

POST /api/v3/projects/{project_id}/clusters/{cluster_id}/nodepools

Table 4-479 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-480 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Table 4-481 Request body parameters

Parameter	Mandatory	Type	Description
kind	Yes	String	API type. The value is fixed at NodePool .
apiVersion	Yes	String	API version. The value is fixed at v3 .
metadata	Yes	NodePoolMetadata object	Metadata information of the node pool
spec	Yes	NodePoolSpec object	Node pool specifications
status	No	NodePoolStatus object	Node pool status

Table 4-482 NodePoolMetadata

Parameter	Mandatory	Type	Description
name	Yes	String	Node pool name. NOTE Naming rules: Enter 1 to 50 characters, starting with a lowercase letter and not ending with a hyphen (-). Only lowercase letters, digits, and hyphens (-) are allowed. <ul style="list-style-type: none"> You cannot create node pools named DefaultPool.
uid	No	String	UID of the node pool. The value is automatically generated after the object is updated. A user-defined value will not take effect.
annotations	No	Map<String,String>	Annotations of a node pool in key-value pairs. This parameter is used only for query and cannot be input through a request. Entered data of this parameter is invalid.
updateTimestamp	No	String	Update time.
creationTimestamp	No	String	Creation time.

Table 4-483 NodePoolSpec

Parameter	Mandatory	Type	Description
type	No	String	Node pool type. If this parameter is left blank, the value vm is used by default. <ul style="list-style-type: none"> vm: ECS ElasticBMS: C6 general computing-plus BMS. An example flavor is c6.22xlarge.2.physical. bms: BMS
nodeTemplate	Yes	NodeSpec object	Detailed parameters of the node pool template.

Parameter	Mandatory	Type	Description
initialNodeCount	No	Integer	Initial number of nodes for the node pool. When queried, the value is the number of target nodes in the node pool.
autoscaling	No	NodePoolNodeAutoscaling object	Auto scaling parameters
nodeManagement	No	NodeManagement object	Node management configuration
podSecurityGroups	No	Array of SecurityID objects	Security group configuration. This parameter is available only for CCE Turbo clusters.
customSecurityGroups	No	Array of strings	<p>Custom security group settings for a node pool. New nodes scaled out in a node pool can be bound to a specified security group.</p> <ul style="list-style-type: none"> • Specifying no security group ID will add the new nodes to the default security group of the worker nodes. • Specifying a valid security group ID will put new nodes in that security group. • When specifying a security group, do not modify the rules of the port on which CCE running depends. For details, see How Do I Harden the Automatically Created Security Group Rules for CCE Cluster Nodes.

Table 4-484 NodeSpec

Parameter	Mandatory	Type	Description
flavor	Yes	String	Node specifications. For details about the node specifications supported by CCE, see Node Flavor Description .

Parameter	Mandatory	Type	Description
az	Yes	String	AZ where the node to be created is located. You need to specify the AZ name. If it is set to random , the node will be created in a random AZ. For details about AZs supported by CCE, see Regions and Endpoints .
os	No	String	Node OS. For details about the supported OSs, see Node OS . NOTE <ul style="list-style-type: none"> The system automatically selects the supported OS based on the cluster version. If the current cluster version does not support the OS, an error will be reported. If alpha.cce/NodeImageID in extendParam is specified during node creation, you do not need to configure this parameter. This parameter is mandatory when creating a node pool. If you are using shared disk space when creating a node, this parameter is mandatory. Specifically, the disk initialization configuration management parameter storage is used and the name field of virtualSpaces in StorageGroups is set to share.
login	Yes	Login object	Node login mode. Either the key pair or password must be used for login.
rootVolume	Yes	Volume object	Information about disks on the node

Parameter	Mandatory	Type	Description
dataVolumes	Yes	Array of Volume objects	<p>Data disk parameters of the node. Currently, you can add the second data disk for your node on the CCE console.</p> <p>This data disk is used by the container runtime and kubelet. Do not uninstall this disk. Otherwise, the node will become unavailable.</p> <p>For DeC nodes, the parameter description is the same as that for rootVolume.</p>
storage	No	Storage object	<p>Disk initialization management parameter.</p> <p>This parameter is complex to configure. For details, see Attaching Disks to a Node.</p> <p>If this parameter retains its default, disks are managed based on the DockerLVMConfigOverride (discarded) parameter in extendParam. This parameter is supported by clusters of version 1.15.11 and later.</p> <p>NOTE</p> <ul style="list-style-type: none"> • If a node specification involves both local disks and EVS disks, do not retain the default value of this parameter to prevent unexpected disk partitions. • If you want to change the value range of a data disk to 20 to 32768, do not use the default value. • If you want to use the shared disk space (with the runtime and Kubernetes partitions cancelled), do not retain the default value of this parameter. For details about the shared disk space, see Data Disk Space Allocation.
publicIP	No	NodePublicIP object	<p>EIP of a node.</p> <p>NOTE</p> <p>This parameter is not supported when you add a node to a node pool.</p>

Parameter	Mandatory	Type	Description
nodeNicSpec	No	NodeNicSpec object	NIC of the node
count	No	Integer	Number of nodes to be created in a batch. The value must be a positive integer greater than or equal to 1 and less than or equal to the defined limit. This parameter can be left blank when it is used for a node pool.
billingMode	No	Integer	Node billing mode. <ul style="list-style-type: none"> • 0: pay-per-use • 1: yearly/monthly • 2: (discarded) yearly/monthly billing with auto payment enabled

Parameter	Mandatory	Type	Description
taints	No	Array of Taint objects	<p>You can add taints to created nodes to configure anti-affinity. A maximum of 20 taints can be added. Each taint contains the following parameters:</p> <ul style="list-style-type: none"> • Key: A key must contain 1 to 63 characters starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain name can be used as the prefix of a key. • Value: A value must start with a letter or digit and can contain a maximum of 63 characters, including letters, digits, hyphens (-), underscores (_), and periods (.). • Effect: Available options are NoSchedule, PreferNoSchedule, and NoExecute. <p>When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query.</p> <p>Example:</p> <pre>"taints": [{ "key": "status", "value": "unavailable", "effect": "NoSchedule" }, { "key": "looks", "value": "bad", "effect": "NoSchedule" }]</pre>

Parameter	Mandatory	Type	Description
k8sTags	No	Map<String,String>	<p>Defined in key-value pairs. A maximum of 20 key-value pairs are allowed.</p> <ul style="list-style-type: none"> • Key: Enter 1 to 63 characters, starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain can be prefixed to a key and contain a maximum of 253 characters. Example DNS subdomain: example.com/my-key • Value: The value can be left blank or contain 1 to 63 characters that start with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed in the character string. <p>When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query.</p> <p>Example:</p> <pre>"k8sTags": { "key": "value" }</pre>
ecsGroupId	No	String	<p>ECS group ID. If this parameter is configured, nodes will be created in the specified ECS group.</p> <p>NOTE This configuration does not take effect when a node pool is created. To ensure that all nodes in a node pool are in the same ECS group, configure nodeManagement in the node pool.</p>

Parameter	Mandatory	Type	Description
dedicatedHostId	No	String	<p>ID of the DeH to which the node is scheduled.</p> <p>NOTE This parameter is not supported when you add a node during node pool creation.</p>
userTags	No	Array of UserTag objects	<p>Cloud server tag. The key of a tag must be unique. The maximum number of custom tags supported by CCE depends on the region and cannot exceed 8.</p> <p>When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query.</p> <p>NOTE Use only letters, Unicode characters, digits, special characters(-, _). Max characters: 36</p>
runtime	No	Runtime object	<p>Container runtime:</p> <ul style="list-style-type: none"> • Clusters of v1.25 or earlier: docker. • Clusters of v1.25 or later: Container runtime varies with the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker. For nodes running other OSs, the default container runtime is containerd.

Parameter	Mandatory	Type	Description
initializedConditions	No	Array of strings	<p>Custom initialization flag, which is left blank by default. Before CCE nodes are initialized, node.cloudprovider.kubernetes.io/uninitialized is added to the nodes to prevent pods from being scheduled to these nodes. When creating a node, you can configure the initializedConditions parameter to control the taint removal time. By default, the timeout period is not configured.</p> <p>Example:</p> <ol style="list-style-type: none"> 1. Create a node and add parameter <code>"initializedConditions": ["CCEInitial", "CustomedInitial"]</code>. 2. After custom initialization is complete, CCE will call a Kubernetes API (for example, PATCH /v1/nodes/{node_ip}/status) to update the node conditions by adding two labels of types CCEInitial and CustomedInitial and setting the status to True, as shown in the following: <pre>status: conditions: - type: CCEInitial status: 'True' - type: CustomedInitial status: 'True'</pre> <ol style="list-style-type: none"> 1. CCE polls status.Conditions of nodes to check whether there are conditions of types CCEInitial and CustomedInitial. If such conditions exist and the status is True, the node initialization is complete and the initialization taint is removed.

Parameter	Mandatory	Type	Description
			<p>2. initializedConditions allows you to configure a timeout period during node creation. For example, "initializedConditions": ["CCEInitial:15m", "CustomedInitial:15m"] indicates that the timeout period is 15 minutes. After the timeout period is reached, the initialization conditions will be automatically ignored and the initialization taint will be removed when CCE polls the node.</p> <ul style="list-style-type: none"> • Use only letters and digits. Max. characters: 20. • Max. flags: 2. • The unit of the timeout period is minute (m).
extendParam	No	NodeExtendParam object	Extended parameters for creating a node.
hostnameConfig	No	HostnameConfig object	Kubernetes node name configuration parameter, which is supported by clusters of v1.23.6-r0 to v1.25 or clusters of v1.25.2-r0 or later versions.

Table 4-485 Login

Parameter	Mandatory	Type	Description
sshKey	No	String	Name of the key pair used for login.
userPassword	No	UserPassword object	Password used for node login.

Table 4-486 UserPassword

Parameter	Mandatory	Type	Description
username	No	String	Login account. The default value is root .
password	Yes	String	<p>If a username and a password are used to create a node, this field is shielded in the response body.</p> <p>A password must meet the following complexity requirements:</p> <ul style="list-style-type: none"> • Contains 8 to 26 characters. • Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (!@\$%^_-=+[{ }];,./?~#*) • Cannot contain the username or the username spelled backwards. <p>The password field must be salted during node creation. For details, see Adding a Salt in the password Field When Creating a Node.</p>

Table 4-487 Volume

Parameter	Mandatory	Type	Description
size	Yes	Integer	<p>Disk size, in GB.</p> <ul style="list-style-type: none"> • System disk: 40 to 1024 • Value range for the first data disk: 20 to 32768 (When storage is not specified, this value range becomes 100 to 32768.) • Value range for other data disks: 10 to 32768. (When storage is not specified, this value range becomes 100 to 32768.)

Parameter	Mandatory	Type	Description
volumetype	Yes	String	Disk type. For details about possible values, see the description of the root_volume parameter in the API used to create an ECS. <ul style="list-style-type: none"> • SAS: high I/O SAS disk • SSD: ultra-high I/O SSD disk • SATA: common I/O SATA disk SATA disks have been removed from EVS. You can find them attached only on existing nodes.
extendParam	No	Map<String,Object>	Extended disk parameters, defined in extendparam in the API used to create an ECS. For details, see Creating an ECS .
cluster_id	No	String	ID of the storage pool used by the ECS system disk. This field is used only for DeC clusters, which functions as dssPoolID , that is, the ID of the DSS storage pool. To obtain the value, see the ID field in the API used to obtain details of a DSS storage pool .
cluster_type	No	String	Storage class of the cloud server system disk. The value is always dss . This field is used only for DeC clusters.

Parameter	Mandatory	Type	Description
hw:passthrough	No	Boolean	<ul style="list-style-type: none"> Pay attention to this field if your ECS is SDI-compliant. If the value of this field is true, an SCSI disk will be created. If the node pool type is ElasticBMS, this field must be set to true. If a node specification involves local disks and EVS disks at the same time, set the disk initialization parameters. For details, see Attaching Disks to a Node.
metadata	No	VolumeMetadata object	EVS disk encryption information. This field is mandatory only when you need to encrypt the system disk or data disks of the node to be created.

Table 4-488 VolumeMetadata

Parameter	Mandatory	Type	Description
__system__encrypted	No	String	<p>Whether the EVS disk is encrypted. The value 0 indicates that the EVS disk is not encrypted, and the value 1 indicates that the EVS disk is encrypted.</p> <p>If this parameter is not specified, EVS disks will not be encrypted by default.</p>
__system__cmkid	No	String	CMK ID, which indicates encryption in metadata . This field is used with __system__encrypted .

Table 4-489 Storage

Parameter	Mandatory	Type	Description
storageSelectors	Yes	Array of StorageSelectors objects	Disk selection. Matched disks are managed according to matchLabels and storageType .
storageGroups	Yes	Array of StorageGroups objects	A storage group consists of multiple storage devices. It is used to divide storage space.

Table 4-490 StorageSelectors

Parameter	Mandatory	Type	Description
name	Yes	String	Selector name, used as the index of selectorNames in storageGroup . Therefore, the name of each selector must be unique.
storageType	Yes	String	Specifies the storage type. Currently, only evs (EVS volumes) and local (local volumes) are supported. The local storage does not support disk selection. All local disks will form a VG. Therefore, only one storageSelector of the local type is allowed.
matchLabels	No	matchLabels object	Matching field of an EVS volume. The size , volumeType , metadataEncrypted , metadataCmkid and count fields are supported.

Table 4-491 matchLabels

Parameter	Mandatory	Type	Description
size	No	String	Matched disk size. If this parameter is left unspecified, the disk size is not limited. Example: 100

Parameter	Mandatory	Type	Description
volumeType	No	String	EVS disk type. SSD , GPSSD , SAS , ESSD , and SATA are supported. If this parameter is left blank, there is no restriction on the disk type.
metadataEncrypted	No	String	Disk encryption identifier. If this parameter is set to 0 , it specifies that the disk is not encrypted. If this parameter is set to 1 , it specifies that the disk is encrypted. If this parameter is left blank, there is no restriction on the disk encryption identifier.
metadataCmkid	No	String	CMK ID of an encrypted disk. The value is a string of 36 bytes. If this parameter is left blank, there is no restriction on the disk key ID.
count	No	String	Number of disks to be selected. If this parameter is left blank, all disks of this type are selected.

Table 4-492 StorageGroups

Parameter	Mandatory	Type	Description
name	Yes	String	Name of a virtual storage group, which must be unique. NOTE <ul style="list-style-type: none"> If cceManaged is set to true, the name must be set to vgpaas. If the data disk is used as a temporary storage volume, the name must be vg-everest-localvolume-ephemeral. If the data disk is used as a persistent storage volume, the name must be vg-everest-localvolume-persistent.

Parameter	Mandatory	Type	Description
cceManaged	No	Boolean	Storage space for Kubernetes and runtime components. Only one group can be set to true . If this parameter is left blank, the default value false is used.
selectorNames	Yes	Array of strings	This parameter corresponds to name in storageSelectors . A group can match multiple selectors, but a selector can match only one group.
virtualSpaces	Yes	Array of VirtualSpace objects	Detailed management of space configuration in a group.

Table 4-493 VirtualSpace

Parameter	Mandatory	Type	Description
name	Yes	String	Name of a virtualSpace. Options: share , kubernetes , runtime , and user <ul style="list-style-type: none"> • share: shared space configuration (with the runtime and Kubernetes partitions cancelled). lvmConfig needs to be configured. • Kubernetes: Kubernetes space configuration. lvmConfig needs to be configured. • runtime: runtime space configuration. runtimeConfig needs to be configured. • user: user space configuration. lvmConfig needs to be configured.
size	Yes	String	Size of a virtualSpace. The value must be an integer in percentage. Example: 90%. NOTE The sum of the percentages of all virtualSpaces in a group cannot exceed 100%.

Parameter	Mandatory	Type	Description
lvmConfig	No	LVMConfig object	LVM configuration managements, applicable to the <i>share*</i> , kubernetes , and user spaces. Note that one virtual space supports only one config.
runtimeConfig	No	RuntimeConfig object	runtime configurations, applicable to the runtime space. Note that one virtual space supports only one config.

Table 4-494 LVMConfig

Parameter	Mandatory	Type	Description
lvType	Yes	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.
path	No	String	Path to which the disk is attached. This parameter takes effect only in user configuration. The value is an absolute path. Digits, letters, periods (.), hyphens (-), and underscores (_) are allowed.

Table 4-495 RuntimeConfig

Parameter	Mandatory	Type	Description
lvType	Yes	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.

Table 4-496 NodePublicIP

Parameter	Mandatory	Type	Description
ids	No	Array of strings	IDs of existing EIPs. The quantity cannot be greater than the number of nodes to be created. NOTE If ids has been set, you do not need to set count and eip .
count	No	Integer	Number of EIPs to be dynamically created. NOTE count and eip must be set at the same time.
eip	No	NodeEIPSpec object	EIP configuration.

Table 4-497 NodeEIPSpec

Parameter	Mandatory	Type	Description
iptype	Yes	String	EIP type, specified in publicip.type in the API for assigning an EIP. For details, see Assigning an EIP .
bandwidth	No	NodeBandwidth object	Bandwidth parameters of the EIP

Table 4-498 NodeBandwidth

Parameter	Mandatory	Type	Description
chargemode	No	String	<p>Bandwidth billing mode.</p> <ul style="list-style-type: none"> If this field is not specified, the billing is based on bandwidth. If the field is null, the billing is based on bandwidth. If the field value is traffic, the billing is based on traffic. If the value is out of the preceding options, the cloud server will fail to be created. <p>NOTE</p> <ul style="list-style-type: none"> Billed by bandwidth: The billing will be based on the data transmission rate (in Mbps) of public networks. This billing mode is recommended if your bandwidth usage is higher than 10%. Billed by traffic: The billing is based on the total amount of data (in GB) transmitted over the public network. This mode is available only when you are creating a pay-per-use node. This billing mode is recommended if your bandwidth usage is lower than 10%.
size	No	Integer	<p>Bandwidth size, specified in bandwidth.size in the API for assigning an EIP.</p> <p>For details, see Assigning an EIP.</p>
sharetype	No	String	<p>Bandwidth sharing type. Value options: PER (exclusive bandwidth)</p>

Table 4-499 NodeNicSpec

Parameter	Mandatory	Type	Description
primaryNic	No	NicSpec object	Description of the primary NIC.
extNics	No	Array of NicSpec objects	Extension NIC NOTE This parameter is not supported when you add a node to a node pool.

Table 4-500 NicSpec

Parameter	Mandatory	Type	Description
subnetId	No	String	Network ID of the subnet to which a NIC belongs. If subnetId is not specified when a primary NIC is creating, the cluster subnet will be used. If subnetList is also configured for a node pool, the subnetList field is used for adding subnets to the node pool. When creating an extension NIC, you must specify subnetId .
fixedIps	No	Array of strings	The IP address of the primary ENI is specified using fixedIps . The number of IP addresses cannot be greater than the number of created nodes. Either fixedIps or ipBlock can be specified. fixedIps cannot be specified for extension ENIs.
ipBlock	No	String	CIDR format of the primary NIC IP range. The IP address of the created node falls in this range. fixedIps and ipBlock cannot be specified at the same time.

Table 4-501 Taint

Parameter	Mandatory	Type	Description
key	Yes	String	Key.
value	No	String	Value.
effect	Yes	String	Effect.

Table 4-502 UserTag

Parameter	Mandatory	Type	Description
key	No	String	Key of the cloud server label. The value cannot start with CCE- or __type_baremetal .
value	No	String	Value of the cloud server label.

Table 4-503 Runtime

Parameter	Mandatory	Type	Description
name	No	String	Container runtime. Default value: <ul style="list-style-type: none"> • Clusters earlier than v1.25: The default value is docker. • Clusters of v1.25 or later: The default container runtime varies depending on the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker; for nodes running other OSs, the default container runtime is containerd.

Table 4-504 NodeExtendParam

Parameter	Mandatory	Type	Description
ecs:performancetype	No	String	ECS flavor types. This field is returned in the response.

Parameter	Mandatory	Type	Description
orderID	No	String	Order ID. This parameter is returned in the response when the node is billed on a yearly/ monthly basis with auto payment enabled (only in creation scenarios).
productID	No	String	Product ID. This field is returned in the response when the node is yearly/monthly-billed with auto payment enabled.
maxPods	No	Integer	Maximum number of pods that can be created on a node, including the default system pods. Value range: 16 to 256. This limit prevents the node from being overloaded of pods. The number of pods that can be created on a node is determined by multiple parameters. For details, see Maximum Number of Pods That Can Be Created on a Node .
periodType	No	String	<ul style="list-style-type: none"> ● month: The unit is month. ● year: The unit is year. <p>NOTE Request parameter, which is valid and mandatory only when billingMode is set to 1 (yearly/ monthly billing) or 2 (yearly/ monthly billing with auto renewal enabled) (discarded). Response parameter, which is returned only when a yearly/ monthly node is created.</p>

Parameter	Mandatory	Type	Description
periodNum	No	Integer	<p>Subscription duration. The value can be:</p> <ul style="list-style-type: none"> • If periodType is month, the value ranges from 1 to 9. • If periodType is year, the value is 1. <p>NOTE Request parameter, which is valid and mandatory when billingMode is set to 1 or 2 (discarded). Response parameter, which is returned only when a yearly/monthly node is created.</p>
isAutoRenew	No	String	<p>Whether auto renewal is enabled.</p> <ul style="list-style-type: none"> • true: Auto renewal is enabled. • false: Auto renewal is not enabled. <p>NOTE This field is valid when billingMode is set to 1 or 2 (discarded). If not specified, auto renewal is not enabled.</p>
isAutoPay	No	String	<p>Whether to deduct fees automatically.</p> <ul style="list-style-type: none"> • true: Enable automatic fee deduction. • false: Do not enable automatic fee deduction. <p>NOTE This field is valid when billingMode is set to 1 or 2 (discarded). If not specified when billingMode is set to 1, auto fee deduction is not enabled. (Discarded: If not specified when billingMode is set to 2, auto fee deduction is enabled.)</p>

Parameter	Mandatory	Type	Description
DockerLVMConfigOverride	No	String	<p>Docker data disk configuration item. (This parameter has been discarded. Use the storage field instead.)</p> <p>Example default configuration: <pre>"DockerLVMConfigOverride": "dockerThinpool=vgpaas/90%VG;kubernetesLV=vgpaas/10%VG;diskType=evs;lvType=linear"</pre></p> <p>By default, if no VD disk is available, an error occurs because the data disk fails to be found. Set diskType based on the actual drive letter type.</p> <p>The following fields are included:</p> <ul style="list-style-type: none"> • userLV (optional): size of the user space, for example, vgpaas/20%VG. • userPath (optional): mount path of the user space, for example, /home/wqt-test. • diskType: disk type. Currently, only evs, hdd, and ssd are supported. • lvType: type of a logic volume. The value can be linear or striped. • dockerThinpool: Docker space size, for example, vgpaas/60%VG. • kubernetesLV: kubelet space size, for example, vgpaas/20%VG.

Parameter	Mandatory	Type	Description
dockerBaseSize	No	Integer	<p>Available disk space of a single container on a node, in GB. This parameter has been discarded. Use containerBaseSize.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In Device Mapper mode, the default value is 10. In OverlayFS mode, the available space of a single container is not limited by default, and the dockerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters of the new versions.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that dockerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device.</p>

Parameter	Mandatory	Type	Description
containerBaseSize	No	Integer	<p>Available disk space of a single container on a node, in GB.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In OverlayFS mode, the available space of a single container is not limited by default. In Device Mapper mode, the default value is 10, and the containerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters v1.23.14-r0, v1.25.9-r0, v1.27.6-r0, v1.28.4-r0 or later.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that containerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device. In new versions, the Device Mapper mode is used only by BMSs in the same resource pool and is being discarded.</p>
publicKey	No	String	Public key of a node.
alpha.cce/preInstall	No	String	<p>Pre-installation script.</p> <p>NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)</p>

Parameter	Mandatory	Type	Description
alpha.cce/postInstall	No	String	Post-installation script. NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)
alpha.cce/NodeImageID	No	String	This parameter is required when a custom image is used to create a BMS node.
nicMultiqueue	No	String	<ul style="list-style-type: none"> Number of ENI queues. Example setting: <code>"[{\"queue\":4}]"</code> The following fields are included: <ul style="list-style-type: none"> queue: number of ENI queues. This field can be configured only for BMS nodes in a CCE Turbo cluster. Supported proportions are {"1":128, "2":92, "4":92, "8":32, "16":16, "28":9}. That is, if there is one queue, a maximum of 128 ENIs can be bound. If there are two queues, a maximum of 92 ENIs can be bound for two queues. A larger number of ENI queues indicates higher performance but fewer ENIs can be bound. The queue settings cannot be changed after the node pool is created.

Parameter	Mandatory	Type	Description
nicThreshold	No	String	<ul style="list-style-type: none"> ENI pre-binding thresholds. Example setting: "0.3:0.6" Low threshold (L): determines the minimum number of pre-bound ENIs (Min). Formula: $Min = Total\ number\ of\ ENIs\ of\ the\ node \times L$ High threshold (H): determines the maximum number of pre-bound ENIs (Max). Formula: $Max = Total\ number\ of\ ENIs\ of\ the\ node \times H$ Number of ENIs bound to a BMS node (B) and number of ENIs being used by pods (U): $U + Min < B < U + Max$ If the number of pre-bound ENIs on a BMS node is smaller than the minimum allowed, the system will bind more ENIs to make the numbers equal. If the number of pre-bound ENIs on a BMS node is larger than the maximum allowed, the system periodically unbinds ENIs (about every 2 minutes) to make the numbers equal. Both the thresholds are one-decimal-place values ranging from 0.0 to 1.0. The low threshold must be smaller than or equal to the high one. This field can be configured only for BMS nodes in a CCE Turbo cluster. Pre-binding ENIs can speed up workload creation but occupies IP addresses.

Parameter	Mandatory	Type	Description
chargingMode	No	Integer	Billing mode of a node. This parameter has been deprecated. Use the billingMode parameter in NodeSpec .
agency_name	No	String	Name of an agency An agency is created by a tenant administrator on Identity and Access Management (IAM) to provide temporary credentials for CCE nodes to access cloud servers. This parameter is returned only when it is transferred during node creation.
kubeReservedMem	No	Integer	Reserved node memory, which is reserved for Kubernetes components.
systemReservedMem	No	Integer	Reserved node memory, which is reserved for system components.
init-node-password	No	String	Node password. If this parameter is used as a response parameter, asterisks (*) will be displayed.

Table 4-505 HostnameConfig

Parameter	Mandatory	Type	Description
type	Yes	String	<p>Configuration type of the Kubernetes node name. The default value is privatelp.</p> <ul style="list-style-type: none"> • privatelp: The Kubernetes node is named after its IP address. • cceNodeName: The Kubernetes node is named after the CCE node. <p>NOTE</p> <ul style="list-style-type: none"> • For a node which is configured using cceNodeName, the name is the same as the Kubernetes node name and the ECS name. The node name cannot be changed. If the ECS name is changed on the ECS console, the node name will retain unchanged after ECS synchronization. • For a node which is configured using cceNodeName, to avoid a conflict between Kubernetes nodes, the system automatically adds a suffix to each node name. The suffix is in the format of A hyphen (-) Five random characters. The value of the random characters is a lowercase letter or a digit ranging from 0 to 9.

Table 4-506 NodePoolNodeAutoscaling

Parameter	Mandatory	Type	Description
enable	No	Boolean	Whether to enable auto scaling.
minNodeCount	No	Integer	Minimum number of nodes allowed if auto scaling is enabled. The value cannot be greater than the maximum number of nodes allowed by the cluster specifications.

Parameter	Mandatory	Type	Description
maxNodeCount	No	Integer	Maximum number of nodes allowed if auto scaling is enabled. This value must be greater than or equal to the value of minNodeCount and cannot exceed the maximum number of nodes in the cluster specifications.
scaleDownCooldownTime	No	Integer	Interval between two scaling operations, in minutes. During this period, nodes added after a scale-up will not be deleted.
priority	No	Integer	Weight of a node pool. A node pool with a higher weight has a higher priority during scaling.

Table 4-507 NodeManagement

Parameter	Mandatory	Type	Description
serverGroupReference	No	String	Cloud server group ID. If this field is specified, all nodes in the node pool will be created in this group. The group ID can be specified only when you create the node pool and cannot be modified. When you specify a cloud server group, the number of nodes in the node pool cannot exceed the group quota.

Table 4-508 SecurityID

Parameter	Mandatory	Type	Description
id	No	String	Security group ID

Table 4-509 NodePoolStatus

Parameter	Mandatory	Type	Description
currentNode	No	Integer	Total number of nodes in the current node pool (excluding the nodes that are being deleted)
creatingNode	No	Integer	Number of nodes in the creation process in the node pool
deletingNode	No	Integer	Number of nodes being deleted in the current node pool.

Parameter	Mandatory	Type	Description
phase	No	String	<p>Node pool status.</p> <ul style="list-style-type: none"> • Null: available (the number of current nodes in the node pool has reached the expected value, and no node scaling is being performed.) • Synchronizing: scaling in progress (the number of current nodes in the node pool does not reach the expected value and no node scaling is being performed.) • Synchronized: pending scaling (the number of current nodes in the node pool does not reach the expected value, or node scaling is being performed.) • SoldOut: The node pool cannot be scaled out. (This field is used in multiple scenarios, for example, resources in the node pool have been sold out or the resource quota is insufficient.) <p>NOTE This node pool status has been discarded and is reserved only for compatibility. You are not advised to use it. The replacements are as follows:</p> <ul style="list-style-type: none"> • Node pool scaling status: You can obtain accurate status of the current node pool based on parameters such as currentNode, creatingNode, or deletingNode. • Node pool scale-out: You can use conditions to obtain the detailed status of a node pool. Scalable can replace SoldOut. • Deleting: The object is being deleted. • Error: An error occurs.

Parameter	Mandatory	Type	Description
jobId	No	String	ID of a job executed on the node pool. This field is available only when the node pool is being deleted.
conditions	No	Array of NodePoolCondition objects	Node pool status details. For details, see the definition of Condition .

Table 4-510 NodePoolCondition

Parameter	Mandatory	Type	Description
type	No	String	<p>Condition type. The options are as follows:</p> <ul style="list-style-type: none"> • Scalable: whether a node pool can be scaled. If the status is False, node pool scaling will not be triggered again. • QuotaInsufficient: Quotas on which node pool scaling depends are insufficient, affecting the node pool scaling status. • ResourceInsufficient: Resources on which node pool scaling depends are insufficient, affecting the node pool scaling status. • UnexpectedError: The node pool fails to be scaled out due to unexpected reasons, affecting the node pool scaling status. • LockedByOrder: The yearly/monthly-billed node pool is locked due to an order. The value of Reason is the ID of the pending order. • Error: A node pool error occurs. A common trigger is deletion failure.

Parameter	Mandatory	Type	Description
status	No	String	Current status of Condition . The options are as follows: <ul style="list-style-type: none"> "True" "False"
lastProbeTime	No	String	Time when the status was last checked
lastTransitTime	No	String	Time when the status was last changed
reason	No	String	Reason why the status was last changed
message	No	String	Detailed condition description

Response Parameters

Status code: 201

Table 4-511 Response body parameters

Parameter	Type	Description
kind	String	API type. The value is fixed at NodePool .
apiVersion	String	API version. The value is fixed at v3 .
metadata	NodePoolMetadata object	Metadata of the node pool.
spec	NodePoolSpec object	Node pool specifications.
status	CreateNodePoolStatus object	Node pool status.

Table 4-512 NodePoolMetadata

Parameter	Type	Description
name	String	Node pool name. NOTE Naming rules: Enter 1 to 50 characters, starting with a lowercase letter and not ending with a hyphen (-). Only lowercase letters, digits, and hyphens (-) are allowed. <ul style="list-style-type: none"> You cannot create node pools named DefaultPool.
uid	String	UID of the node pool. The value is automatically generated after the object is updated. A user-defined value will not take effect.
annotations	Map<String,String>	Annotations of a node pool in key-value pairs. This parameter is used only for query and cannot be input through a request. Entered data of this parameter is invalid.
updateTimestamp	String	Update time.
creationTimestamp	String	Creation time.

Table 4-513 NodePoolSpec

Parameter	Type	Description
type	String	Node pool type. If this parameter is left blank, the value vm is used by default. <ul style="list-style-type: none"> vm: ECS ElasticBMS: C6 general computing-plus BMS. An example flavor is c6.22xlarge.2.physical. bms: BMS
nodeTemplate	NodeSpec object	Detailed parameters of the node pool template.
initialNodeCount	Integer	Initial number of nodes for the node pool. When queried, the value is the number of target nodes in the node pool.
autoscaling	NodePoolNodeAutoscaling object	Auto scaling parameters

Parameter	Type	Description
nodeManagement	NodeManagement object	Node management configuration
podSecurityGroups	Array of SecurityID objects	Security group configuration. This parameter is available only for CCE Turbo clusters.
customSecurityGroups	Array of strings	<p>Custom security group settings for a node pool. New nodes scaled out in a node pool can be bound to a specified security group.</p> <ul style="list-style-type: none"> • Specifying no security group ID will add the new nodes to the default security group of the worker nodes. • Specifying a valid security group ID will put new nodes in that security group. • When specifying a security group, do not modify the rules of the port on which CCE running depends. For details, see How Do I Harden the Automatically Created Security Group Rules for CCE Cluster Nodes.

Table 4-514 NodeSpec

Parameter	Type	Description
flavor	String	Node specifications. For details about the node specifications supported by CCE, see Node Flavor Description .
az	String	<p>AZ where the node to be created is located. You need to specify the AZ name. If it is set to random, the node will be created in a random AZ.</p> <p>For details about AZs supported by CCE, see Regions and Endpoints.</p>

Parameter	Type	Description
os	String	<p>Node OS. For details about the supported OSs, see Node OS.</p> <p>NOTE</p> <ul style="list-style-type: none"> The system automatically selects the supported OS based on the cluster version. If the current cluster version does not support the OS, an error will be reported. If <code>alpha.cce/NodeImageID</code> in <code>extendParam</code> is specified during node creation, you do not need to configure this parameter. This parameter is mandatory when creating a node pool. If you are using shared disk space when creating a node, this parameter is mandatory. Specifically, the disk initialization configuration management parameter <code>storage</code> is used and the <code>name</code> field of <code>virtualSpaces</code> in <code>StorageGroups</code> is set to <code>share</code>.
login	Login object	Node login mode. Either the key pair or password must be used for login.
rootVolume	Volume object	Information about disks on the node
dataVolumes	Array of Volume objects	<p>Data disk parameters of the node. Currently, you can add the second data disk for your node on the CCE console.</p> <p>This data disk is used by the container runtime and kubelet. Do not uninstall this disk. Otherwise, the node will become unavailable.</p> <p>For DeC nodes, the parameter description is the same as that for <code>rootVolume</code>.</p>

Parameter	Type	Description
storage	Storage object	<p>Disk initialization management parameter. This parameter is complex to configure. For details, see Attaching Disks to a Node.</p> <p>If this parameter retains its default, disks are managed based on the DockerLVMConfigOverride (discarded) parameter in extendParam. This parameter is supported by clusters of version 1.15.11 and later.</p> <p>NOTE</p> <ul style="list-style-type: none"> • If a node specification involves both local disks and EVS disks, do not retain the default value of this parameter to prevent unexpected disk partitions. • If you want to change the value range of a data disk to 20 to 32768, do not use the default value. • If you want to use the shared disk space (with the runtime and Kubernetes partitions cancelled), do not retain the default value of this parameter. For details about the shared disk space, see Data Disk Space Allocation.
publicIP	NodePublicIP object	<p>EIP of a node.</p> <p>NOTE This parameter is not supported when you add a node to a node pool.</p>
nodeNicSpec	NodeNicSpec object	NIC of the node
count	Integer	Number of nodes to be created in a batch. The value must be a positive integer greater than or equal to 1 and less than or equal to the defined limit. This parameter can be left blank when it is used for a node pool.
billingMode	Integer	<p>Node billing mode.</p> <ul style="list-style-type: none"> • 0: pay-per-use • 1: yearly/monthly • 2: (discarded) yearly/monthly billing with auto payment enabled

Parameter	Type	Description
taints	Array of Taint objects	<p>You can add taints to created nodes to configure anti-affinity. A maximum of 20 taints can be added. Each taint contains the following parameters:</p> <ul style="list-style-type: none"> • Key: A key must contain 1 to 63 characters starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain name can be used as the prefix of a key. • Value: A value must start with a letter or digit and can contain a maximum of 63 characters, including letters, digits, hyphens (-), underscores (_), and periods (.). • Effect: Available options are NoSchedule, PreferNoSchedule, and NoExecute. When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query. <p>Example:</p> <pre data-bbox="815 1108 1430 1339">"taints": [{ "key": "status", "value": "unavailable", "effect": "NoSchedule" }, { "key": "looks", "value": "bad", "effect": "NoSchedule" }]</pre>

Parameter	Type	Description
k8sTags	Map<String,String>	<p>Defined in key-value pairs. A maximum of 20 key-value pairs are allowed.</p> <ul style="list-style-type: none"> Key: Enter 1 to 63 characters, starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain can be prefixed to a key and contain a maximum of 253 characters. Example DNS subdomain: example.com/my-key Value: The value can be left blank or contain 1 to 63 characters that start with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed in the character string. When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query. <p>Example:</p> <pre>"k8sTags": { "key": "value" }</pre>
ecsGroupId	String	<p>ECS group ID. If this parameter is configured, nodes will be created in the specified ECS group.</p> <p>NOTE This configuration does not take effect when a node pool is created. To ensure that all nodes in a node pool are in the same ECS group, configure nodeManagement in the node pool.</p>
dedicatedHostId	String	<p>ID of the DeH to which the node is scheduled.</p> <p>NOTE This parameter is not supported when you add a node during node pool creation.</p>

Parameter	Type	Description
userTags	Array of UserTag objects	<p>Cloud server tag. The key of a tag must be unique. The maximum number of custom tags supported by CCE depends on the region and cannot exceed 8.</p> <p>When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query.</p> <p>NOTE Use only letters, Unicode characters, digits, special characters(-, _). Max characters: 36</p>
runtime	Runtime object	<p>Container runtime:</p> <ul style="list-style-type: none"> • Clusters of v1.25 or earlier: docker. • Clusters of v1.25 or later: Container runtime varies with the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker. For nodes running other OSs, the default container runtime is containerd.

Parameter	Type	Description
initializedConditions	Array of strings	<p>Custom initialization flag, which is left blank by default.</p> <p>Before CCE nodes are initialized, node.cloudprovider.kubernetes.io/uninitialized is added to the nodes to prevent pods from being scheduled to these nodes. When creating a node, you can configure the initializedConditions parameter to control the taint removal time. By default, the timeout period is not configured.</p> <p>Example:</p> <ol style="list-style-type: none"> 1. Create a node and add parameter "initializedConditions": ["CCEInitial", "CustomedInitial"]. 2. After custom initialization is complete, CCE will call a Kubernetes API (for example, PATCH /v1/nodes/{node_ip}/status) to update the node conditions by adding two labels of types CCEInitial and CustomedInitial and setting the status to True, as shown in the following: <pre data-bbox="815 1055 1430 1205">status: conditions: - type: CCEInitial status: 'True' - type: CustomedInitial status: 'True'</pre> <ol style="list-style-type: none"> 1. CCE polls status.Conditions of nodes to check whether there are conditions of types CCEInitial and CustomedInitial. If such conditions exist and the status is True, the node initialization is complete and the initialization taint is removed. 2. initializedConditions allows you to configure a timeout period during node creation. For example, "initializedConditions": ["CCEInitial:15m", "CustomedInitial:15m"] indicates that the timeout period is 15 minutes. After the timeout period is reached, the initialization conditions will be automatically ignored and the initialization taint will be removed when CCE polls the node. <ul style="list-style-type: none"> ● Use only letters and digits. Max. characters: 20. ● Max. flags: 2. ● The unit of the timeout period is minute (m).

Parameter	Type	Description
extendParam	NodeExtendParam object	Extended parameters for creating a node.
hostnameConfig	HostnameConfig object	Kubernetes node name configuration parameter, which is supported by clusters of v1.23.6-r0 to v1.25 or clusters of v1.25.2-r0 or later versions.

Table 4-515 Login

Parameter	Type	Description
sshKey	String	Name of the key pair used for login.
userPassword	UserPassword object	Password used for node login.

Table 4-516 UserPassword

Parameter	Type	Description
username	String	Login account. The default value is root .
password	String	<p>If a username and a password are used to create a node, this field is shielded in the response body.</p> <p>A password must meet the following complexity requirements:</p> <ul style="list-style-type: none"> • Contains 8 to 26 characters. • Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (!@\$%^-_=+[{ } ; , / ? ~ # *) • Cannot contain the username or the username spelled backwards. <p>The password field must be salted during node creation. For details, see Adding a Salt in the password Field When Creating a Node.</p>

Table 4-517 Volume

Parameter	Type	Description
size	Integer	Disk size, in GB. <ul style="list-style-type: none"> • System disk: 40 to 1024 • Value range for the first data disk: 20 to 32768 (When storage is not specified, this value range becomes 100 to 32768.) • Value range for other data disks: 10 to 32768. (When storage is not specified, this value range becomes 100 to 32768.)
volumetype	String	Disk type. For details about possible values, see the description of the root_volume parameter in the API used to create an ECS. <ul style="list-style-type: none"> • SAS: high I/O SAS disk • SSD: ultra-high I/O SSD disk • SATA: common I/O SATA disk SATA disks have been removed from EVS. You can find them attached only on existing nodes.
extendParam	Map<String, Object>	Extended disk parameters, defined in extendparam in the API used to create an ECS. For details, see Creating an ECS .
cluster_id	String	ID of the storage pool used by the ECS system disk. This field is used only for DeC clusters, which functions as dssPoolID , that is, the ID of the DSS storage pool. To obtain the value, see the ID field in the API used to obtain details of a DSS storage pool .
cluster_type	String	Storage class of the cloud server system disk. The value is always dss . This field is used only for DeC clusters.
hw:passthrough	Boolean	<ul style="list-style-type: none"> • Pay attention to this field if your ECS is SDI-compliant. If the value of this field is true, an SCSI disk will be created. • If the node pool type is ElasticBMS, this field must be set to true. • If a node specification involves local disks and EVS disks at the same time, set the disk initialization parameters. For details, see Attaching Disks to a Node.
metadata	VolumeMetadata object	EVS disk encryption information. This field is mandatory only when you need to encrypt the system disk or data disks of the node to be created.

Table 4-518 VolumeMetadata

Parameter	Type	Description
__system__encrypted	String	Whether the EVS disk is encrypted. The value 0 indicates that the EVS disk is not encrypted, and the value 1 indicates that the EVS disk is encrypted. If this parameter is not specified, EVS disks will not be encrypted by default.
__system__cmkid	String	CMK ID, which indicates encryption in metadata . This field is used with __system__encrypted .

Table 4-519 Storage

Parameter	Type	Description
storageSelectors	Array of StorageSelectors objects	Disk selection. Matched disks are managed according to matchLabels and storageType .
storageGroups	Array of StorageGroups objects	A storage group consists of multiple storage devices. It is used to divide storage space.

Table 4-520 StorageSelectors

Parameter	Type	Description
name	String	Selector name, used as the index of selectorNames in storageGroup . Therefore, the name of each selector must be unique.
storageType	String	Specifies the storage type. Currently, only evs (EVS volumes) and local (local volumes) are supported. The local storage does not support disk selection. All local disks will form a VG. Therefore, only one storageSelector of the local type is allowed.
matchLabels	matchLabels object	Matching field of an EVS volume. The size , volumeType , metadataEncrypted , metadataCmkid and count fields are supported.

Table 4-521 matchLabels

Parameter	Type	Description
size	String	Matched disk size. If this parameter is left unspecified, the disk size is not limited. Example: 100
volumeType	String	EVS disk type. SSD, GPSSD, SAS, ESSD, and SATA are supported. If this parameter is left blank, there is no restriction on the disk type.
metadataEncrypted	String	Disk encryption identifier. If this parameter is set to 0 , it specifies that the disk is not encrypted. If this parameter is set to 1 , it specifies that the disk is encrypted. If this parameter is left blank, there is no restriction on the disk encryption identifier.
metadataCmkid	String	CMK ID of an encrypted disk. The value is a string of 36 bytes. If this parameter is left blank, there is no restriction on the disk key ID.
count	String	Number of disks to be selected. If this parameter is left blank, all disks of this type are selected.

Table 4-522 StorageGroups

Parameter	Type	Description
name	String	Name of a virtual storage group, which must be unique. NOTE <ul style="list-style-type: none"> If cceManaged is set to true, the name must be set to vgpaas. If the data disk is used as a temporary storage volume, the name must be vg-everest-localvolume-ephemeral. If the data disk is used as a persistent storage volume, the name must be vg-everest-localvolume-persistent.
cceManaged	Boolean	Storage space for Kubernetes and runtime components. Only one group can be set to true . If this parameter is left blank, the default value false is used.
selectorNames	Array of strings	This parameter corresponds to name in storageSelectors . A group can match multiple selectors, but a selector can match only one group.

Parameter	Type	Description
virtualSpaces	Array of VirtualSpace objects	Detailed management of space configuration in a group.

Table 4-523 VirtualSpace

Parameter	Type	Description
name	String	Name of a virtualSpace. Options: share , kubernetes , runtime , and user <ul style="list-style-type: none"> • share: shared space configuration (with the runtime and Kubernetes partitions cancelled). lvmConfig needs to be configured. • Kubernetes: Kubernetes space configuration. lvmConfig needs to be configured. • runtime: runtime space configuration. runtimeConfig needs to be configured. • user: user space configuration. lvmConfig needs to be configured.
size	String	Size of a virtualSpace. The value must be an integer in percentage. Example: 90%. NOTE The sum of the percentages of all virtualSpaces in a group cannot exceed 100%.
lvmConfig	LVMConfig object	LVM configuration managements, applicable to the <i>share</i> *, kubernetes , and user spaces. Note that one virtual space supports only one config.
runtimeConfig	RuntimeConfig object	runtime configurations, applicable to the runtime space. Note that one virtual space supports only one config.

Table 4-524 LVMConfig

Parameter	Type	Description
lvType	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.

Parameter	Type	Description
path	String	Path to which the disk is attached. This parameter takes effect only in user configuration. The value is an absolute path. Digits, letters, periods (.), hyphens (-), and underscores (_) are allowed.

Table 4-525 RuntimeConfig

Parameter	Type	Description
lvType	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.

Table 4-526 NodePublicIP

Parameter	Type	Description
ids	Array of strings	IDs of existing EIPs. The quantity cannot be greater than the number of nodes to be created. NOTE If ids has been set, you do not need to set count and eip .
count	Integer	Number of EIPs to be dynamically created. NOTE count and eip must be set at the same time.
eip	NodeEIPSpec object	EIP configuration.

Table 4-527 NodeEIPSpec

Parameter	Type	Description
iptype	String	EIP type, specified in publicip.type in the API for assigning an EIP. For details, see Assigning an EIP .
bandwidth	NodeBandwidth object	Bandwidth parameters of the EIP

Table 4-528 NodeBandwidth

Parameter	Type	Description
chargemode	String	<p>Bandwidth billing mode.</p> <ul style="list-style-type: none"> • If this field is not specified, the billing is based on bandwidth. • If the field is null, the billing is based on bandwidth. • If the field value is traffic, the billing is based on traffic. • If the value is out of the preceding options, the cloud server will fail to be created. <p>NOTE</p> <ul style="list-style-type: none"> • Billed by bandwidth: The billing will be based on the data transmission rate (in Mbps) of public networks. This billing mode is recommended if your bandwidth usage is higher than 10%. • Billed by traffic: The billing is based on the total amount of data (in GB) transmitted over the public network. This mode is available only when you are creating a pay-per-use node. This billing mode is recommended if your bandwidth usage is lower than 10%.
size	Integer	<p>Bandwidth size, specified in bandwidth.size in the API for assigning an EIP.</p> <p>For details, see Assigning an EIP.</p>
sharetype	String	<p>Bandwidth sharing type. Value options: PER (exclusive bandwidth)</p>

Table 4-529 NodeNicSpec

Parameter	Type	Description
primaryNic	NicSpec object	Description of the primary NIC.
extNics	Array of NicSpec objects	<p>Extension NIC</p> <p>NOTE This parameter is not supported when you add a node to a node pool.</p>

Table 4-530 NicSpec

Parameter	Type	Description
subnetId	String	Network ID of the subnet to which a NIC belongs. If subnetId is not specified when a primary NIC is creating, the cluster subnet will be used. If subnetList is also configured for a node pool, the subnetList field is used for adding subnets to the node pool. When creating an extension NIC, you must specify subnetId .
fixedIps	Array of strings	The IP address of the primary ENI is specified using fixedIps . The number of IP addresses cannot be greater than the number of created nodes. Either fixedIps or ipBlock can be specified. fixedIps cannot be specified for extension ENIs.
ipBlock	String	CIDR format of the primary NIC IP range. The IP address of the created node falls in this range. fixedIps and ipBlock cannot be specified at the same time.

Table 4-531 Taint

Parameter	Type	Description
key	String	Key.
value	String	Value.
effect	String	Effect.

Table 4-532 UserTag

Parameter	Type	Description
key	String	Key of the cloud server label. The value cannot start with CCE- or __type_baremetal .
value	String	Value of the cloud server label.

Table 4-533 Runtime

Parameter	Type	Description
name	String	<p>Container runtime. Default value:</p> <ul style="list-style-type: none"> • Clusters earlier than v1.25: The default value is docker. • Clusters of v1.25 or later: The default container runtime varies depending on the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker; for nodes running other OSs, the default container runtime is containerd.

Table 4-534 NodeExtendParam

Parameter	Type	Description
ecs:performancetype	String	ECS flavor types. This field is returned in the response.
orderId	String	Order ID. This parameter is returned in the response when the node is billed on a yearly/monthly basis with auto payment enabled (only in creation scenarios).
productId	String	Product ID. This field is returned in the response when the node is yearly/monthly-billed with auto payment enabled.
maxPods	Integer	<p>Maximum number of pods that can be created on a node, including the default system pods. Value range: 16 to 256.</p> <p>This limit prevents the node from being overloaded of pods.</p> <p>The number of pods that can be created on a node is determined by multiple parameters. For details, see Maximum Number of Pods That Can Be Created on a Node.</p>
periodType	String	<ul style="list-style-type: none"> • month: The unit is month. • year: The unit is year. <p>NOTE</p> <p>Request parameter, which is valid and mandatory only when billingMode is set to 1 (yearly/monthly billing) or 2 (yearly/monthly billing with auto renewal enabled) (discarded).</p> <p>Response parameter, which is returned only when a yearly/monthly node is created.</p>

Parameter	Type	Description
periodNum	Integer	<p>Subscription duration. The value can be:</p> <ul style="list-style-type: none"> • If periodType is month, the value ranges from 1 to 9. • If periodType is year, the value is 1. <p>NOTE Request parameter, which is valid and mandatory when billingMode is set to 1 or 2 (discarded). Response parameter, which is returned only when a yearly/monthly node is created.</p>
isAutoRenew	String	<p>Whether auto renewal is enabled.</p> <ul style="list-style-type: none"> • true: Auto renewal is enabled. • false: Auto renewal is not enabled. <p>NOTE This field is valid when billingMode is set to 1 or 2 (discarded). If not specified, auto renewal is not enabled.</p>
isAutoPay	String	<p>Whether to deduct fees automatically.</p> <ul style="list-style-type: none"> • true: Enable automatic fee deduction. • false: Do not enable automatic fee deduction. <p>NOTE This field is valid when billingMode is set to 1 or 2 (discarded). If not specified when billingMode is set to 1, auto fee deduction is not enabled. (Discarded: If not specified when billingMode is set to 2, auto fee deduction is enabled.)</p>

Parameter	Type	Description
DockerLVMConfigOverride	String	<p>Docker data disk configuration item. (This parameter has been discarded. Use the storage field instead.) Example default configuration: <code>"DockerLVMConfigOverride":"dockerThinpool=vgpaas/90%VG;kubernetesLV=vgpaas/10%VG;diskType=evs;lvType=linear"</code></p> <p>By default, if no VD disk is available, an error occurs because the data disk fails to be found. Set diskType based on the actual drive letter type.</p> <p>The following fields are included:</p> <ul style="list-style-type: none"> • userLV (optional): size of the user space, for example, vgpaas/20%VG. • userPath (optional): mount path of the user space, for example, /home/wqt-test. • diskType: disk type. Currently, only evs, hdd, and ssd are supported. • lvType: type of a logic volume. The value can be linear or striped. • dockerThinpool: Docker space size, for example, vgpaas/60%VG. • kubernetesLV: kubelet space size, for example, vgpaas/20%VG.
dockerBaseSize	Integer	<p>Available disk space of a single container on a node, in GB. This parameter has been discarded. Use containerBaseSize.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In Device Mapper mode, the default value is 10. In OverlayFS mode, the available space of a single container is not limited by default, and the dockerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters of the new versions.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that dockerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device.</p>

Parameter	Type	Description
containerBaseSize	Integer	<p>Available disk space of a single container on a node, in GB.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In OverlayFS mode, the available space of a single container is not limited by default. In Device Mapper mode, the default value is 10, and the containerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters v1.23.14-r0, v1.25.9-r0, v1.27.6-r0, v1.28.4-r0 or later.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that containerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device. In new versions, the Device Mapper mode is used only by BMSs in the same resource pool and is being discarded.</p>
publicKey	String	Public key of a node.
alpha.cce/preInstall	String	<p>Pre-installation script.</p> <p>NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)</p>
alpha.cce/postInstall	String	<p>Post-installation script.</p> <p>NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)</p>
alpha.cce/NodeImageID	String	This parameter is required when a custom image is used to create a BMS node.

Parameter	Type	Description
nicMultiqueue	String	<ul style="list-style-type: none"> Number of ENI queues. Example setting: <code>"[{"queue":4}]"</code> The following fields are included: <ul style="list-style-type: none"> queue: number of ENI queues. This field can be configured only for BMS nodes in a CCE Turbo cluster. Supported proportions are {"1":128, "2":92, "4":92, "8":32, "16":16, "28":9}. That is, if there is one queue, a maximum of 128 ENIs can be bound. If there are two queues, a maximum of 92 ENIs can be bound for two queues. A larger number of ENI queues indicates higher performance but fewer ENIs can be bound. The queue settings cannot be changed after the node pool is created.
nicThreshold	String	<ul style="list-style-type: none"> ENI pre-binding thresholds. Example setting: <code>"0.3:0.6"</code> <ul style="list-style-type: none"> Low threshold (L): determines the minimum number of pre-bound ENIs (Min). Formula: $Min = Total\ number\ of\ ENIs\ of\ the\ node \times L$ High threshold (H): determines the maximum number of pre-bound ENIs (Max). Formula: $Max = Total\ number\ of\ ENIs\ of\ the\ node \times H$ Number of ENIs bound to a BMS node (B) and number of ENIs being used by pods (U): $U + Min < B < U + Max$ If the number of pre-bound ENIs on a BMS node is smaller than the minimum allowed, the system will bind more ENIs to make the numbers equal. If the number of pre-bound ENIs on a BMS node is larger than the maximum allowed, the system periodically unbinds ENIs (about every 2 minutes) to make the numbers equal. Both the thresholds are one-decimal-place values ranging from 0.0 to 1.0. The low threshold must be smaller than or equal to the high one. This field can be configured only for BMS nodes in a CCE Turbo cluster. Pre-binding ENIs can speed up workload creation but occupies IP addresses.

Parameter	Type	Description
chargingMode	Integer	Billing mode of a node. This parameter has been deprecated. Use the billingMode parameter in NodeSpec .
agency_name	String	Name of an agency An agency is created by a tenant administrator on Identity and Access Management (IAM) to provide temporary credentials for CCE nodes to access cloud servers. This parameter is returned only when it is transferred during node creation.
kubeReservedMem	Integer	Reserved node memory, which is reserved for Kubernetes components.
systemReservedMem	Integer	Reserved node memory, which is reserved for system components.
init-node-password	String	Node password. If this parameter is used as a response parameter, asterisks (*) will be displayed.

Table 4-535 HostnameConfig

Parameter	Type	Description
type	String	Configuration type of the Kubernetes node name. The default value is privatelp . <ul style="list-style-type: none"> • privatelp: The Kubernetes node is named after its IP address. • cceNodeName: The Kubernetes node is named after the CCE node. <p>NOTE</p> <ul style="list-style-type: none"> • For a node which is configured using cceNodeName, the name is the same as the Kubernetes node name and the ECS name. The node name cannot be changed. If the ECS name is changed on the ECS console, the node name will retain unchanged after ECS synchronization. • For a node which is configured using cceNodeName, to avoid a conflict between Kubernetes nodes, the system automatically adds a suffix to each node name. The suffix is in the format of A hyphen (-) Five random characters. The value of the random characters is a lowercase letter or a digit ranging from 0 to 9.

Table 4-536 NodePoolNodeAutoscaling

Parameter	Type	Description
enable	Boolean	Whether to enable auto scaling.
minNodeCount	Integer	Minimum number of nodes allowed if auto scaling is enabled. The value cannot be greater than the maximum number of nodes allowed by the cluster specifications.
maxNodeCount	Integer	Maximum number of nodes allowed if auto scaling is enabled. This value must be greater than or equal to the value of minNodeCount and cannot exceed the maximum number of nodes in the cluster specifications.
scaleDownCooldownTime	Integer	Interval between two scaling operations, in minutes. During this period, nodes added after a scale-up will not be deleted.
priority	Integer	Weight of a node pool. A node pool with a higher weight has a higher priority during scaling.

Table 4-537 NodeManagement

Parameter	Type	Description
serverGroupReference	String	Cloud server group ID. If this field is specified, all nodes in the node pool will be created in this group. The group ID can be specified only when you create the node pool and cannot be modified. When you specify a cloud server group, the number of nodes in the node pool cannot exceed the group quota.

Table 4-538 SecurityID

Parameter	Type	Description
id	String	Security group ID

Table 4-539 CreateNodePoolStatus

Parameter	Type	Description
currentNode	Integer	Total number of nodes in the current node pool (excluding the nodes that are being deleted).

Parameter	Type	Description
creatingNode	Integer	Number of nodes that are being created in the node pool.
deletingNode	Integer	Number of nodes that are being deleted in the current node pool.
phase	String	<p>Node pool status. Options:</p> <ul style="list-style-type: none"> • Null: The node pool is available. (The number of nodes in the current node pool has reached the expected value, and no node scaling is being performed.) • Synchronizing: The node pool is being scaled. (The number of nodes in the current node pool has not reached the expected value, and no node scaling is being performed.) • Synchronized: The node pool scaling is pending. (The number of nodes in the current node pool has not reached the expected value, or node scaling is being performed.) • SoldOut: The node pool cannot be scaled out. (This field is used in multiple scenarios, for example, resources in the node pool have been sold out or the resource quota is insufficient.) <p>NOTE This node pool status has been deprecated. It is reserved only for compatibility. Do not use it anymore. Use the following instead:</p> <ul style="list-style-type: none"> • Node pool scaling status: You can obtain the status of the current node pool using parameters such as currentNode, creatingNode, or deletingNode. • Node pool scale-out: You can use conditions to obtain the detailed status of a node pool. Scalable can replace SoldOut. • Deleting: The node pool is being deleted. • Error: An error occurred in the node pool.
conditions	Array of NodePoolCondition objects	Node pool status details. For details, see the definition of Condition .

Table 4-540 NodePoolCondition

Parameter	Type	Description
type	String	Condition type. The options are as follows: <ul style="list-style-type: none"> ● Scalable: whether a node pool can be scaled. If the status is False, node pool scaling will not be triggered again. ● QuotaInsufficient: Quotas on which node pool scaling depends are insufficient, affecting the node pool scaling status. ● ResourceInsufficient: Resources on which node pool scaling depends are insufficient, affecting the node pool scaling status. ● UnexpectedError: The node pool fails to be scaled out due to unexpected reasons, affecting the node pool scaling status. ● LockedByOrder: The yearly/monthly-billed node pool is locked due to an order. The value of Reason is the ID of the pending order. ● Error: A node pool error occurs. A common trigger is deletion failure.
status	String	Current status of Condition . The options are as follows: <ul style="list-style-type: none"> ● "True" ● "False"
lastProbeTime	String	Time when the status was last checked
lastTransitTime	String	Time when the status was last changed
reason	String	Reason why the status was last changed
message	String	Detailed condition description

Example Requests

Create a node pool billed on a pay-per-use basis with the following configurations: The number of nodes is 0. The node pool flavor is 2 vCPUs and 4 GiB of memory. The node OS is EulerOS 2.5. Docker containers are deployed. The sizes of the system disk and data disk of the node are 40 GB and 100 GB, respectively. Both the disks are of the high I/O type.

```
{
  "kind": "NodePool",
  "apiVersion": "v3",
  "metadata": {
    "name": "lc-it-nodepool-79796"
  },
  "spec": {
```

```
"initialNodeCount" : 0,
"type" : "vm",
"autoscaling" : {
  "enable" : false,
  "minNodeCount" : 0,
  "maxNodeCount" : 1,
  "scaleDownCooldownTime" : 0,
  "priority" : 0
},
"nodeManagement" : {
  "serverGroupReference" : ""
},
"nodeTemplate" : {
  "flavor" : "s6.large.2",
  "az" : "*****",
  "os" : "EulerOS 2.5",
  "login" : {
    "sshKey" : "KeyPair-001"
  },
  "rootVolume" : {
    "volumetype" : "SAS",
    "size" : 40
  },
  "dataVolumes" : [ {
    "volumetype" : "SAS",
    "size" : 100,
    "extendParam" : {
      "useType" : "docker"
    }
  } ],
  "billingMode" : 0,
  "extendParam" : {
    "alpha.cce/preInstall" : "",
    "alpha.cce/postInstall" : "",
    "alpha.cce/NodeImageID" : "",
    "maxPods" : 110
  },
  "nodeNicSpec" : {
    "primaryNic" : {
      "subnetId" : "7e767d10-7548-4df5-ad72-aeac1d08bd8a"
    }
  },
  "podSecurityGroups" : [ {
    "id" : ""
  } ]
}
```

Example Responses

Status code: 201

The job for creating a node pool in a specified cluster is successfully delivered.

```
{
  "kind" : "NodePool",
  "apiVersion" : "v3",
  "metadata" : {
    "name" : "lc-it-nodepool-79796",
    "uid" : "99addaa2-69eb-11ea-a592-0255ac1001bb"
  },
  "spec" : {
    "type" : "vm",
    "nodeTemplate" : {
      "flavor" : "s6.large.2",
      "az" : "*****",
      "os" : "EulerOS 2.5",
      "login" : {
```

```

"sshKey" : "KeyPair-001"
},
"rootVolume" : {
  "volumetype" : "SAS",
  "size" : 40
},
"dataVolumes" : [ {
  "volumetype" : "SAS",
  "size" : 100,
  "extendParam" : {
    "useType" : "docker"
  }
} ],
"publicIP" : {
  "eip" : {
    "bandwidth" : { }
  }
},
"nodeNicSpec" : {
  "primaryNic" : {
    "subnetId" : "7e767d10-7548-4df5-ad72-aeac1d08bd8a"
  }
},
"billingMode" : 0,
"extendParam" : {
  "alpha.cce/NodeImageID" : "",
  "alpha.cce/postInstall" : "",
  "alpha.cce/preInstall" : "",
  "maxPods" : 110
},
"k8sTags" : {
  "cce.cloud.com/cce-nodepool" : "lc-it-nodepool-79796"
}
},
"autoscaling" : {
  "maxNodeCount" : 1
},
"nodeManagement" : { }
},
"status" : {
  "phase" : ""
}
}

```

Status Codes

Status Code	Description
201	The job for creating a node pool in a specified cluster is successfully delivered.

Error Codes

See [Error Codes](#).

4.4.2 Reading a Specified Node Pool

Function

This API is used to obtain details about a specified node pool.

 NOTE

The URL for cluster management is in the format of https://Endpoint/uri. In the URL, **uri** indicates the resource path, that is, the path for API access.

Calling Method

For details, see [Calling APIs](#).

URI

GET /api/v3/projects/{project_id}/clusters/{cluster_id}/nodepools/{nodepool_id}

Table 4-541 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
nodepool_id	Yes	String	Node pool ID.

Request Parameters

Table 4-542 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-543 Response body parameters

Parameter	Type	Description
kind	String	API type. The value is fixed at NodePool .
apiVersion	String	API version. The value is fixed at v3 .
metadata	NodePoolMetadata object	Metadata of the node pool.
spec	NodePoolSpec object	Node pool specifications.
status	NodePoolStatus object	Node pool status.

Table 4-544 NodePoolMetadata

Parameter	Type	Description
name	String	Node pool name. NOTE Naming rules: Enter 1 to 50 characters, starting with a lowercase letter and not ending with a hyphen (-). Only lowercase letters, digits, and hyphens (-) are allowed. <ul style="list-style-type: none"> You cannot create node pools named DefaultPool.
uid	String	UID of the node pool. The value is automatically generated after the object is updated. A user-defined value will not take effect.
annotations	Map<String,String>	Annotations of a node pool in key-value pairs. This parameter is used only for query and cannot be input through a request. Entered data of this parameter is invalid.
updateTimestamp	String	Update time.
creationTimestamp	String	Creation time.

Table 4-545 NodePoolSpec

Parameter	Type	Description
type	String	Node pool type. If this parameter is left blank, the value vm is used by default. <ul style="list-style-type: none"> • vm: ECS • ElasticBMS: C6 general computing-plus BMS. An example flavor is c6.22xlarge.2.physical. • bms: BMS
nodeTemplate	NodeSpec object	Detailed parameters of the node pool template.
initialNodeCount	Integer	Initial number of nodes for the node pool. When queried, the value is the number of target nodes in the node pool.
autoscaling	NodePoolNodeAutoscaling object	Auto scaling parameters
nodeManagement	NodeManagement object	Node management configuration
podSecurityGroups	Array of SecurityID objects	Security group configuration. This parameter is available only for CCE Turbo clusters.
customSecurityGroups	Array of strings	Custom security group settings for a node pool. New nodes scaled out in a node pool can be bound to a specified security group. <ul style="list-style-type: none"> • Specifying no security group ID will add the new nodes to the default security group of the worker nodes. • Specifying a valid security group ID will put new nodes in that security group. • When specifying a security group, do not modify the rules of the port on which CCE running depends. For details, see How Do I Harden the Automatically Created Security Group Rules for CCE Cluster Nodes.

Table 4-546 NodeSpec

Parameter	Type	Description
flavor	String	Node specifications. For details about the node specifications supported by CCE, see Node Flavor Description .
az	String	AZ where the node to be created is located. You need to specify the AZ name. If it is set to random , the node will be created in a random AZ. For details about AZs supported by CCE, see Regions and Endpoints .
os	String	Node OS. For details about the supported OSs, see Node OS . NOTE <ul style="list-style-type: none"> The system automatically selects the supported OS based on the cluster version. If the current cluster version does not support the OS, an error will be reported. If alpha.cce/NodeImageID in extendParam is specified during node creation, you do not need to configure this parameter. This parameter is mandatory when creating a node pool. If you are using shared disk space when creating a node, this parameter is mandatory. Specifically, the disk initialization configuration management parameter storage is used and the name field of virtualSpaces in StorageGroups is set to share.
login	Login object	Node login mode. Either the key pair or password must be used for login.
rootVolume	Volume object	Information about disks on the node
dataVolumes	Array of Volume objects	Data disk parameters of the node. Currently, you can add the second data disk for your node on the CCE console. This data disk is used by the container runtime and kubelet. Do not uninstall this disk. Otherwise, the node will become unavailable. For DeC nodes, the parameter description is the same as that for rootVolume .

Parameter	Type	Description
storage	Storage object	<p>Disk initialization management parameter. This parameter is complex to configure. For details, see Attaching Disks to a Node.</p> <p>If this parameter retains its default, disks are managed based on the DockerLVMConfigOverride (discarded) parameter in extendParam. This parameter is supported by clusters of version 1.15.11 and later.</p> <p>NOTE</p> <ul style="list-style-type: none"> • If a node specification involves both local disks and EVS disks, do not retain the default value of this parameter to prevent unexpected disk partitions. • If you want to change the value range of a data disk to 20 to 32768, do not use the default value. • If you want to use the shared disk space (with the runtime and Kubernetes partitions cancelled), do not retain the default value of this parameter. For details about the shared disk space, see Data Disk Space Allocation.
publicIP	NodePublicIP object	<p>EIP of a node.</p> <p>NOTE This parameter is not supported when you add a node to a node pool.</p>
nodeNicSpec	NodeNicSpec object	NIC of the node
count	Integer	Number of nodes to be created in a batch. The value must be a positive integer greater than or equal to 1 and less than or equal to the defined limit. This parameter can be left blank when it is used for a node pool.
billingMode	Integer	<p>Node billing mode.</p> <ul style="list-style-type: none"> • 0: pay-per-use • 1: yearly/monthly • 2: (discarded) yearly/monthly billing with auto payment enabled

Parameter	Type	Description
taints	Array of Taint objects	<p>You can add taints to created nodes to configure anti-affinity. A maximum of 20 taints can be added. Each taint contains the following parameters:</p> <ul style="list-style-type: none"> • Key: A key must contain 1 to 63 characters starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain name can be used as the prefix of a key. • Value: A value must start with a letter or digit and can contain a maximum of 63 characters, including letters, digits, hyphens (-), underscores (_), and periods (.). • Effect: Available options are NoSchedule, PreferNoSchedule, and NoExecute. When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query. <p>Example:</p> <pre>"taints": [{ "key": "status", "value": "unavailable", "effect": "NoSchedule" }, { "key": "looks", "value": "bad", "effect": "NoSchedule" }]</pre>

Parameter	Type	Description
k8sTags	Map<String,String>	<p>Defined in key-value pairs. A maximum of 20 key-value pairs are allowed.</p> <ul style="list-style-type: none"> Key: Enter 1 to 63 characters, starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain can be prefixed to a key and contain a maximum of 253 characters. Example DNS subdomain: example.com/my-key Value: The value can be left blank or contain 1 to 63 characters that start with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed in the character string. When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query. <p>Example:</p> <pre>"k8sTags": { "key": "value" }</pre>
ecsGroupId	String	<p>ECS group ID. If this parameter is configured, nodes will be created in the specified ECS group.</p> <p>NOTE This configuration does not take effect when a node pool is created. To ensure that all nodes in a node pool are in the same ECS group, configure nodeManagement in the node pool.</p>
dedicatedHostId	String	<p>ID of the DeH to which the node is scheduled.</p> <p>NOTE This parameter is not supported when you add a node during node pool creation.</p>

Parameter	Type	Description
userTags	Array of UserTag objects	<p>Cloud server tag. The key of a tag must be unique. The maximum number of custom tags supported by CCE depends on the region and cannot exceed 8.</p> <p>When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query.</p> <p>NOTE Use only letters, Unicode characters, digits, special characters(-, _). Max characters: 36</p>
runtime	Runtime object	<p>Container runtime:</p> <ul style="list-style-type: none"> • Clusters of v1.25 or earlier: docker. • Clusters of v1.25 or later: Container runtime varies with the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker. For nodes running other OSs, the default container runtime is containerd.

Parameter	Type	Description
initializedConditions	Array of strings	<p>Custom initialization flag, which is left blank by default.</p> <p>Before CCE nodes are initialized, node.cloudprovider.kubernetes.io/uninitialized is added to the nodes to prevent pods from being scheduled to these nodes. When creating a node, you can configure the initializedConditions parameter to control the taint removal time. By default, the timeout period is not configured.</p> <p>Example:</p> <ol style="list-style-type: none"> 1. Create a node and add parameter "initializedConditions": ["CCEInitial", "CustomedInitial"]. 2. After custom initialization is complete, CCE will call a Kubernetes API (for example, PATCH /v1/nodes/{node_ip}/status) to update the node conditions by adding two labels of types CCEInitial and CustomedInitial and setting the status to True, as shown in the following: <pre data-bbox="815 1055 1430 1205"> status: conditions: - type: CCEInitial status: 'True' - type: CustomedInitial status: 'True' </pre> <ol style="list-style-type: none"> 1. CCE polls status.Conditions of nodes to check whether there are conditions of types CCEInitial and CustomedInitial. If such conditions exist and the status is True, the node initialization is complete and the initialization taint is removed. 2. initializedConditions allows you to configure a timeout period during node creation. For example, "initializedConditions": ["CCEInitial:15m", "CustomedInitial:15m"] indicates that the timeout period is 15 minutes. After the timeout period is reached, the initialization conditions will be automatically ignored and the initialization taint will be removed when CCE polls the node. <ul style="list-style-type: none"> ● Use only letters and digits. Max. characters: 20. ● Max. flags: 2. ● The unit of the timeout period is minute (m).

Parameter	Type	Description
extendParam	NodeExtendParam object	Extended parameters for creating a node.
hostnameConfig	HostnameConfig object	Kubernetes node name configuration parameter, which is supported by clusters of v1.23.6-r0 to v1.25 or clusters of v1.25.2-r0 or later versions.

Table 4-547 Login

Parameter	Type	Description
sshKey	String	Name of the key pair used for login.
userPassword	UserPassword object	Password used for node login.

Table 4-548 UserPassword

Parameter	Type	Description
username	String	Login account. The default value is root .
password	String	<p>If a username and a password are used to create a node, this field is shielded in the response body.</p> <p>A password must meet the following complexity requirements:</p> <ul style="list-style-type: none"> • Contains 8 to 26 characters. • Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (!@\$%^-_=+[{ }];,./?~#*) • Cannot contain the username or the username spelled backwards. <p>The password field must be salted during node creation. For details, see Adding a Salt in the password Field When Creating a Node.</p>

Table 4-549 Volume

Parameter	Type	Description
size	Integer	Disk size, in GB. <ul style="list-style-type: none"> • System disk: 40 to 1024 • Value range for the first data disk: 20 to 32768 (When storage is not specified, this value range becomes 100 to 32768.) • Value range for other data disks: 10 to 32768. (When storage is not specified, this value range becomes 100 to 32768.)
volumetype	String	Disk type. For details about possible values, see the description of the root_volume parameter in the API used to create an ECS. <ul style="list-style-type: none"> • SAS: high I/O SAS disk • SSD: ultra-high I/O SSD disk • SATA: common I/O SATA disk SATA disks have been removed from EVS. You can find them attached only on existing nodes.
extendParam	Map<String, Object>	Extended disk parameters, defined in extendparam in the API used to create an ECS. For details, see Creating an ECS .
cluster_id	String	ID of the storage pool used by the ECS system disk. This field is used only for DeC clusters, which functions as dssPoolID , that is, the ID of the DSS storage pool. To obtain the value, see the ID field in the API used to obtain details of a DSS storage pool .
cluster_type	String	Storage class of the cloud server system disk. The value is always dss . This field is used only for DeC clusters.
hw:passthrough	Boolean	<ul style="list-style-type: none"> • Pay attention to this field if your ECS is SDI-compliant. If the value of this field is true, an SCSI disk will be created. • If the node pool type is ElasticBMS, this field must be set to true. • If a node specification involves local disks and EVS disks at the same time, set the disk initialization parameters. For details, see Attaching Disks to a Node.
metadata	VolumeMetadata object	EVS disk encryption information. This field is mandatory only when you need to encrypt the system disk or data disks of the node to be created.

Table 4-550 VolumeMetadata

Parameter	Type	Description
__system__encrypted	String	Whether the EVS disk is encrypted. The value 0 indicates that the EVS disk is not encrypted, and the value 1 indicates that the EVS disk is encrypted. If this parameter is not specified, EVS disks will not be encrypted by default.
__system__cmkid	String	CMK ID, which indicates encryption in metadata . This field is used with __system__encrypted .

Table 4-551 Storage

Parameter	Type	Description
storageSelectors	Array of StorageSelectors objects	Disk selection. Matched disks are managed according to matchLabels and storageType .
storageGroups	Array of StorageGroups objects	A storage group consists of multiple storage devices. It is used to divide storage space.

Table 4-552 StorageSelectors

Parameter	Type	Description
name	String	Selector name, used as the index of selectorNames in storageGroup . Therefore, the name of each selector must be unique.
storageType	String	Specifies the storage type. Currently, only evs (EVS volumes) and local (local volumes) are supported. The local storage does not support disk selection. All local disks will form a VG. Therefore, only one storageSelector of the local type is allowed.
matchLabels	matchLabels object	Matching field of an EVS volume. The size , volumeType , metadataEncrypted , metadataCmkid and count fields are supported.

Table 4-553 matchLabels

Parameter	Type	Description
size	String	Matched disk size. If this parameter is left unspecified, the disk size is not limited. Example: 100
volumeType	String	EVS disk type. SSD, GPSSD, SAS, ESSD, and SATA are supported. If this parameter is left blank, there is no restriction on the disk type.
metadataEncrypted	String	Disk encryption identifier. If this parameter is set to 0 , it specifies that the disk is not encrypted. If this parameter is set to 1 , it specifies that the disk is encrypted. If this parameter is left blank, there is no restriction on the disk encryption identifier.
metadataCmkid	String	CMK ID of an encrypted disk. The value is a string of 36 bytes. If this parameter is left blank, there is no restriction on the disk key ID.
count	String	Number of disks to be selected. If this parameter is left blank, all disks of this type are selected.

Table 4-554 StorageGroups

Parameter	Type	Description
name	String	Name of a virtual storage group, which must be unique. NOTE <ul style="list-style-type: none"> If cceManaged is set to true, the name must be set to vgpaas. If the data disk is used as a temporary storage volume, the name must be vg-everest-localvolume-ephemeral. If the data disk is used as a persistent storage volume, the name must be vg-everest-localvolume-persistent.
cceManaged	Boolean	Storage space for Kubernetes and runtime components. Only one group can be set to true . If this parameter is left blank, the default value false is used.
selectorNames	Array of strings	This parameter corresponds to name in storageSelectors . A group can match multiple selectors, but a selector can match only one group.

Parameter	Type	Description
virtualSpaces	Array of VirtualSpace objects	Detailed management of space configuration in a group.

Table 4-555 VirtualSpace

Parameter	Type	Description
name	String	Name of a virtualSpace. Options: share , kubernetes , runtime , and user <ul style="list-style-type: none"> • share: shared space configuration (with the runtime and Kubernetes partitions cancelled). lvmConfig needs to be configured. • Kubernetes: Kubernetes space configuration. lvmConfig needs to be configured. • runtime: runtime space configuration. runtimeConfig needs to be configured. • user: user space configuration. lvmConfig needs to be configured.
size	String	Size of a virtualSpace. The value must be an integer in percentage. Example: 90%. NOTE The sum of the percentages of all virtualSpaces in a group cannot exceed 100%.
lvmConfig	LVMConfig object	LVM configuration managements, applicable to the <i>share</i> *, kubernetes , and user spaces. Note that one virtual space supports only one config.
runtimeConfig	RuntimeConfig object	runtime configurations, applicable to the runtime space. Note that one virtual space supports only one config.

Table 4-556 LVMConfig

Parameter	Type	Description
lvType	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.

Parameter	Type	Description
path	String	Path to which the disk is attached. This parameter takes effect only in user configuration. The value is an absolute path. Digits, letters, periods (.), hyphens (-), and underscores (_) are allowed.

Table 4-557 RuntimeConfig

Parameter	Type	Description
lvType	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.

Table 4-558 NodePublicIP

Parameter	Type	Description
ids	Array of strings	IDs of existing EIPs. The quantity cannot be greater than the number of nodes to be created. NOTE If ids has been set, you do not need to set count and eip .
count	Integer	Number of EIPs to be dynamically created. NOTE count and eip must be set at the same time.
eip	NodeEIPSpec object	EIP configuration.

Table 4-559 NodeEIPSpec

Parameter	Type	Description
iptype	String	EIP type, specified in publicip.type in the API for assigning an EIP. For details, see Assigning an EIP .
bandwidth	NodeBandwidth object	Bandwidth parameters of the EIP

Table 4-560 NodeBandwidth

Parameter	Type	Description
chargemode	String	<p>Bandwidth billing mode.</p> <ul style="list-style-type: none"> • If this field is not specified, the billing is based on bandwidth. • If the field is null, the billing is based on bandwidth. • If the field value is traffic, the billing is based on traffic. • If the value is out of the preceding options, the cloud server will fail to be created. <p>NOTE</p> <ul style="list-style-type: none"> • Billed by bandwidth: The billing will be based on the data transmission rate (in Mbps) of public networks. This billing mode is recommended if your bandwidth usage is higher than 10%. • Billed by traffic: The billing is based on the total amount of data (in GB) transmitted over the public network. This mode is available only when you are creating a pay-per-use node. This billing mode is recommended if your bandwidth usage is lower than 10%.
size	Integer	<p>Bandwidth size, specified in bandwidth.size in the API for assigning an EIP.</p> <p>For details, see Assigning an EIP.</p>
sharetype	String	<p>Bandwidth sharing type. Value options: PER (exclusive bandwidth)</p>

Table 4-561 NodeNicSpec

Parameter	Type	Description
primaryNic	NicSpec object	Description of the primary NIC.
extNics	Array of NicSpec objects	<p>Extension NIC</p> <p>NOTE This parameter is not supported when you add a node to a node pool.</p>

Table 4-562 NicSpec

Parameter	Type	Description
subnetId	String	Network ID of the subnet to which a NIC belongs. If subnetId is not specified when a primary NIC is creating, the cluster subnet will be used. If subnetList is also configured for a node pool, the subnetList field is used for adding subnets to the node pool. When creating an extension NIC, you must specify subnetId .
fixedIps	Array of strings	The IP address of the primary ENI is specified using fixedIps . The number of IP addresses cannot be greater than the number of created nodes. Either fixedIps or ipBlock can be specified. fixedIps cannot be specified for extension ENIs.
ipBlock	String	CIDR format of the primary NIC IP range. The IP address of the created node falls in this range. fixedIps and ipBlock cannot be specified at the same time.

Table 4-563 Taint

Parameter	Type	Description
key	String	Key.
value	String	Value.
effect	String	Effect.

Table 4-564 UserTag

Parameter	Type	Description
key	String	Key of the cloud server label. The value cannot start with CCE- or __type_baremetal .
value	String	Value of the cloud server label.

Table 4-565 Runtime

Parameter	Type	Description
name	String	<p>Container runtime. Default value:</p> <ul style="list-style-type: none"> • Clusters earlier than v1.25: The default value is docker. • Clusters of v1.25 or later: The default container runtime varies depending on the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker; for nodes running other OSs, the default container runtime is containerd.

Table 4-566 NodeExtendParam

Parameter	Type	Description
ecs:performancetype	String	ECS flavor types. This field is returned in the response.
orderId	String	Order ID. This parameter is returned in the response when the node is billed on a yearly/monthly basis with auto payment enabled (only in creation scenarios).
productId	String	Product ID. This field is returned in the response when the node is yearly/monthly-billed with auto payment enabled.
maxPods	Integer	<p>Maximum number of pods that can be created on a node, including the default system pods. Value range: 16 to 256.</p> <p>This limit prevents the node from being overloaded of pods.</p> <p>The number of pods that can be created on a node is determined by multiple parameters. For details, see Maximum Number of Pods That Can Be Created on a Node.</p>
periodType	String	<ul style="list-style-type: none"> • month: The unit is month. • year: The unit is year. <p>NOTE</p> <p>Request parameter, which is valid and mandatory only when billingMode is set to 1 (yearly/monthly billing) or 2 (yearly/monthly billing with auto renewal enabled) (discarded).</p> <p>Response parameter, which is returned only when a yearly/monthly node is created.</p>

Parameter	Type	Description
periodNum	Integer	<p>Subscription duration. The value can be:</p> <ul style="list-style-type: none"> • If periodType is month, the value ranges from 1 to 9. • If periodType is year, the value is 1. <p>NOTE Request parameter, which is valid and mandatory when billingMode is set to 1 or 2 (discarded). Response parameter, which is returned only when a yearly/monthly node is created.</p>
isAutoRenew	String	<p>Whether auto renewal is enabled.</p> <ul style="list-style-type: none"> • true: Auto renewal is enabled. • false: Auto renewal is not enabled. <p>NOTE This field is valid when billingMode is set to 1 or 2 (discarded). If not specified, auto renewal is not enabled.</p>
isAutoPay	String	<p>Whether to deduct fees automatically.</p> <ul style="list-style-type: none"> • true: Enable automatic fee deduction. • false: Do not enable automatic fee deduction. <p>NOTE This field is valid when billingMode is set to 1 or 2 (discarded). If not specified when billingMode is set to 1, auto fee deduction is not enabled. (Discarded: If not specified when billingMode is set to 2, auto fee deduction is enabled.)</p>

Parameter	Type	Description
DockerLVMConfigOverride	String	<p>Docker data disk configuration item. (This parameter has been discarded. Use the storage field instead.) Example default configuration: "DockerLVMConfigOverride":"dockerThinpool=vgpaas/90%VG;kubernetesLV=vgpaas/10%VG;diskType=evs;lvType=linear"</p> <p>By default, if no VD disk is available, an error occurs because the data disk fails to be found. Set diskType based on the actual drive letter type.</p> <p>The following fields are included:</p> <ul style="list-style-type: none"> • userLV (optional): size of the user space, for example, vgpaas/20%VG. • userPath (optional): mount path of the user space, for example, /home/wqt-test. • diskType: disk type. Currently, only evs, hdd, and ssd are supported. • lvType: type of a logic volume. The value can be linear or striped. • dockerThinpool: Docker space size, for example, vgpaas/60%VG. • kubernetesLV: kubelet space size, for example, vgpaas/20%VG.
dockerBaseSize	Integer	<p>Available disk space of a single container on a node, in GB. This parameter has been discarded. Use containerBaseSize.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In Device Mapper mode, the default value is 10. In OverlayFS mode, the available space of a single container is not limited by default, and the dockerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters of the new versions.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that dockerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device.</p>

Parameter	Type	Description
containerBaseSize	Integer	<p>Available disk space of a single container on a node, in GB.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In OverlayFS mode, the available space of a single container is not limited by default. In Device Mapper mode, the default value is 10, and the containerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters v1.23.14-r0, v1.25.9-r0, v1.27.6-r0, v1.28.4-r0 or later.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that containerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device. In new versions, the Device Mapper mode is used only by BMSs in the same resource pool and is being discarded.</p>
publicKey	String	Public key of a node.
alpha.cce/preInstall	String	<p>Pre-installation script.</p> <p>NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)</p>
alpha.cce/postInstall	String	<p>Post-installation script.</p> <p>NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)</p>
alpha.cce/NodeImageID	String	This parameter is required when a custom image is used to create a BMS node.

Parameter	Type	Description
nicMultiqueue	String	<ul style="list-style-type: none"> Number of ENI queues. Example setting: <code>"[{"queue":4}]"</code> The following fields are included: <ul style="list-style-type: none"> queue: number of ENI queues. This field can be configured only for BMS nodes in a CCE Turbo cluster. Supported proportions are {"1":128, "2":92, "4":92, "8":32, "16":16, "28":9}. That is, if there is one queue, a maximum of 128 ENIs can be bound. If there are two queues, a maximum of 92 ENIs can be bound for two queues. A larger number of ENI queues indicates higher performance but fewer ENIs can be bound. The queue settings cannot be changed after the node pool is created.
nicThreshold	String	<ul style="list-style-type: none"> ENI pre-binding thresholds. Example setting: <code>"0.3:0.6"</code> <ul style="list-style-type: none"> Low threshold (L): determines the minimum number of pre-bound ENIs (Min). Formula: $Min = Total\ number\ of\ ENIs\ of\ the\ node \times L$ High threshold (H): determines the maximum number of pre-bound ENIs (Max). Formula: $Max = Total\ number\ of\ ENIs\ of\ the\ node \times H$ Number of ENIs bound to a BMS node (B) and number of ENIs being used by pods (U): $U + Min < B < U + Max$ If the number of pre-bound ENIs on a BMS node is smaller than the minimum allowed, the system will bind more ENIs to make the numbers equal. If the number of pre-bound ENIs on a BMS node is larger than the maximum allowed, the system periodically unbinds ENIs (about every 2 minutes) to make the numbers equal. Both the thresholds are one-decimal-place values ranging from 0.0 to 1.0. The low threshold must be smaller than or equal to the high one. This field can be configured only for BMS nodes in a CCE Turbo cluster. Pre-binding ENIs can speed up workload creation but occupies IP addresses.

Parameter	Type	Description
chargingMode	Integer	Billing mode of a node. This parameter has been deprecated. Use the billingMode parameter in NodeSpec .
agency_name	String	Name of an agency An agency is created by a tenant administrator on Identity and Access Management (IAM) to provide temporary credentials for CCE nodes to access cloud servers. This parameter is returned only when it is transferred during node creation.
kubeReservedMem	Integer	Reserved node memory, which is reserved for Kubernetes components.
systemReservedMem	Integer	Reserved node memory, which is reserved for system components.
init-node-password	String	Node password. If this parameter is used as a response parameter, asterisks (*) will be displayed.

Table 4-567 HostnameConfig

Parameter	Type	Description
type	String	Configuration type of the Kubernetes node name. The default value is privatelp . <ul style="list-style-type: none"> • privatelp: The Kubernetes node is named after its IP address. • cceNodeName: The Kubernetes node is named after the CCE node. <p>NOTE</p> <ul style="list-style-type: none"> • For a node which is configured using cceNodeName, the name is the same as the Kubernetes node name and the ECS name. The node name cannot be changed. If the ECS name is changed on the ECS console, the node name will retain unchanged after ECS synchronization. • For a node which is configured using cceNodeName, to avoid a conflict between Kubernetes nodes, the system automatically adds a suffix to each node name. The suffix is in the format of A hyphen (-) Five random characters. The value of the random characters is a lowercase letter or a digit ranging from 0 to 9.

Table 4-568 NodePoolNodeAutoscaling

Parameter	Type	Description
enable	Boolean	Whether to enable auto scaling.
minNodeCount	Integer	Minimum number of nodes allowed if auto scaling is enabled. The value cannot be greater than the maximum number of nodes allowed by the cluster specifications.
maxNodeCount	Integer	Maximum number of nodes allowed if auto scaling is enabled. This value must be greater than or equal to the value of minNodeCount and cannot exceed the maximum number of nodes in the cluster specifications.
scaleDownCooldownTime	Integer	Interval between two scaling operations, in minutes. During this period, nodes added after a scale-up will not be deleted.
priority	Integer	Weight of a node pool. A node pool with a higher weight has a higher priority during scaling.

Table 4-569 NodeManagement

Parameter	Type	Description
serverGroupReference	String	Cloud server group ID. If this field is specified, all nodes in the node pool will be created in this group. The group ID can be specified only when you create the node pool and cannot be modified. When you specify a cloud server group, the number of nodes in the node pool cannot exceed the group quota.

Table 4-570 SecurityID

Parameter	Type	Description
id	String	Security group ID

Table 4-571 NodePoolStatus

Parameter	Type	Description
currentNode	Integer	Total number of nodes in the current node pool (excluding the nodes that are being deleted)

Parameter	Type	Description
creatingNode	Integer	Number of nodes in the creation process in the node pool
deletingNode	Integer	Number of nodes being deleted in the current node pool.
phase	String	<p>Node pool status.</p> <ul style="list-style-type: none"> • Null: available (the number of current nodes in the node pool has reached the expected value, and no node scaling is being performed.) • Synchronizing: scaling in progress (the number of current nodes in the node pool does not reach the expected value and no node scaling is being performed.) • Synchronized: pending scaling (the number of current nodes in the node pool does not reach the expected value, or node scaling is being performed.) • SoldOut: The node pool cannot be scaled out. (This field is used in multiple scenarios, for example, resources in the node pool have been sold out or the resource quota is insufficient.) <p>NOTE This node pool status has been discarded and is reserved only for compatibility. You are not advised to use it. The replacements are as follows:</p> <ul style="list-style-type: none"> • Node pool scaling status: You can obtain accurate status of the current node pool based on parameters such as currentNode, creatingNode, or deletingNode. • Node pool scale-out: You can use conditions to obtain the detailed status of a node pool. Scalable can replace SoldOut. • Deleting: The object is being deleted. • Error: An error occurs.
jobId	String	ID of a job executed on the node pool. This field is available only when the node pool is being deleted.
conditions	Array of NodePoolCondition objects	Node pool status details. For details, see the definition of Condition .

Table 4-572 NodePoolCondition

Parameter	Type	Description
type	String	Condition type. The options are as follows: <ul style="list-style-type: none"> ● Scalable: whether a node pool can be scaled. If the status is False, node pool scaling will not be triggered again. ● QuotaInsufficient: Quotas on which node pool scaling depends are insufficient, affecting the node pool scaling status. ● ResourceInsufficient: Resources on which node pool scaling depends are insufficient, affecting the node pool scaling status. ● UnexpectedError: The node pool fails to be scaled out due to unexpected reasons, affecting the node pool scaling status. ● LockedByOrder: The yearly/monthly-billed node pool is locked due to an order. The value of Reason is the ID of the pending order. ● Error: A node pool error occurs. A common trigger is deletion failure.
status	String	Current status of Condition . The options are as follows: <ul style="list-style-type: none"> ● "True" ● "False"
lastProbeTime	String	Time when the status was last checked
lastTransitTime	String	Time when the status was last changed
reason	String	Reason why the status was last changed
message	String	Detailed condition description

Example Requests

None

Example Responses

Status code: 200

Details about the specified node pool are obtained successfully.

```
{
  "kind": "NodePool",
  "apiVersion": "v3",
  "metadata": {
    "name": "lc-it-nodepool-79796",
```

```
"uid" : "99addaa2-69eb-11ea-a592-0255ac1001bb"
},
"spec" : {
  "type" : "vm",
  "nodeTemplate" : {
    "flavor" : "s6.large.2",
    "az" : "*****",
    "os" : "EulerOS 2.5",
    "login" : {
      "sshKey" : "KeyPair-001"
    }
  },
  "rootVolume" : {
    "volumetype" : "SAS",
    "size" : 40
  },
  "dataVolumes" : [ {
    "volumetype" : "SAS",
    "size" : 100,
    "extendParam" : {
      "useType" : "docker"
    }
  } ],
  "publicIP" : {
    "eip" : {
      "bandwidth" : { }
    }
  },
  "nodeNicSpec" : {
    "primaryNic" : {
      "subnetId" : "7e767d10-7548-4df5-ad72-aeac1d08bd8a"
    }
  },
  "billingMode" : 0,
  "extendParam" : {
    "maxPods" : 110
  },
  "k8sTags" : {
    "cce.cloud.com/cce-nodepool" : "lc-it-nodepool-79796"
  }
},
"autoscaling" : { },
"nodeManagement" : { }
},
"status" : {
  "phase" : "Deleting",
  "jobId" : "3281fa02-69ee-11ea-a592-0255ac1001bb"
}
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ShowNodePoolSolution {
```

```
public static void main(String[] args) {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running
    // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    String ak = System.getenv("CLOUD_SDK_AK");
    String sk = System.getenv("CLOUD_SDK_SK");
    String projectId = "{project_id}";

    ICredential auth = new BasicCredentials()
        .withProjectId(projectId)
        .withAk(ak)
        .withSk(sk);

    CceClient client = CceClient.newBuilder()
        .withCredential(auth)
        .withRegion(CceRegion.valueOf("<YOUR REGION>"))
        .build();
    ShowNodePoolRequest request = new ShowNodePoolRequest();
    request.withClusterId("{cluster_id}");
    request.withNodepoolId("{nodepool_id}");
    try {
        ShowNodePoolResponse response = client.showNodePool(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowNodePoolRequest()
        request.cluster_id = "{cluster_id}"
```

```
request.nodepool_id = "{nodepool_id}"
response = client.show_node_pool(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowNodePoolRequest{}
    request.ClusterId = "{cluster_id}"
    request.NodepoolId = "{nodepool_id}"
    response, err := client.ShowNodePool(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Details about the specified node pool are obtained successfully.

Error Codes

See [Error Codes](#).

4.4.3 Listing All Node Pools in a Specified Cluster

Function

This API is used to obtain information about all node pools in a specified cluster.

NOTE

- The URL for cluster management is in the format of **https://Endpoint/uri**. In the URL, **uri** indicates the resource path, that is, the path for API access.
- A node pool is a group of nodes within a cluster that all have the same configuration.

Calling Method

For details, see [Calling APIs](#).

URI

GET /api/v3/projects/{project_id}/clusters/{cluster_id}/nodepools

Table 4-573 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Table 4-574 Query Parameters

Parameter	Mandatory	Type	Description
showDefaultNodePool	No	String	Whether to display the default node pool. By default, the default node pool is not displayed. If this parameter is set to true , the default node pool is displayed.

Request Parameters

Table 4-575 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-576 Response body parameters

Parameter	Type	Description
kind	String	API type. The value is fixed to List.
apiVersion	String	API version. The value is fixed to v3.
items	Array of NodePoolResp objects	/

Table 4-577 NodePoolResp

Parameter	Type	Description
kind	String	API type. The value is fixed at NodePool .
apiVersion	String	API version. The value is fixed at v3 .
metadata	NodePoolMetadata object	Metadata of the node pool.
spec	NodePoolSpec object	Node pool specifications.
status	NodePoolStatus object	Node pool status.

Table 4-578 NodePoolMetadata

Parameter	Type	Description
name	String	Node pool name. NOTE Naming rules: Enter 1 to 50 characters, starting with a lowercase letter and not ending with a hyphen (-). Only lowercase letters, digits, and hyphens (-) are allowed. <ul style="list-style-type: none"> You cannot create node pools named DefaultPool.
uid	String	UID of the node pool. The value is automatically generated after the object is updated. A user-defined value will not take effect.
annotations	Map<String,String>	Annotations of a node pool in key-value pairs. This parameter is used only for query and cannot be input through a request. Entered data of this parameter is invalid.
updateTimestamp	String	Update time.
creationTimestamp	String	Creation time.

Table 4-579 NodePoolSpec

Parameter	Type	Description
type	String	Node pool type. If this parameter is left blank, the value vm is used by default. <ul style="list-style-type: none"> vm: ECS ElasticBMS: C6 general computing-plus BMS. An example flavor is c6.22xlarge.2.physical. bms: BMS
nodeTemplate	NodeSpec object	Detailed parameters of the node pool template.
initialNodeCount	Integer	Initial number of nodes for the node pool. When queried, the value is the number of target nodes in the node pool.
autoscaling	NodePoolNodeAutoscaling object	Auto scaling parameters

Parameter	Type	Description
nodeManagement	NodeManagement object	Node management configuration
podSecurityGroups	Array of SecurityID objects	Security group configuration. This parameter is available only for CCE Turbo clusters.
customSecurityGroups	Array of strings	<p>Custom security group settings for a node pool. New nodes scaled out in a node pool can be bound to a specified security group.</p> <ul style="list-style-type: none"> • Specifying no security group ID will add the new nodes to the default security group of the worker nodes. • Specifying a valid security group ID will put new nodes in that security group. • When specifying a security group, do not modify the rules of the port on which CCE running depends. For details, see How Do I Harden the Automatically Created Security Group Rules for CCE Cluster Nodes.

Table 4-580 NodeSpec

Parameter	Type	Description
flavor	String	Node specifications. For details about the node specifications supported by CCE, see Node Flavor Description .
az	String	<p>AZ where the node to be created is located. You need to specify the AZ name. If it is set to random, the node will be created in a random AZ.</p> <p>For details about AZs supported by CCE, see Regions and Endpoints.</p>

Parameter	Type	Description
os	String	<p>Node OS. For details about the supported OSs, see Node OS.</p> <p>NOTE</p> <ul style="list-style-type: none"> The system automatically selects the supported OS based on the cluster version. If the current cluster version does not support the OS, an error will be reported. If <code>alpha.cce/NodeImageID</code> in <code>extendParam</code> is specified during node creation, you do not need to configure this parameter. This parameter is mandatory when creating a node pool. If you are using shared disk space when creating a node, this parameter is mandatory. Specifically, the disk initialization configuration management parameter <code>storage</code> is used and the <code>name</code> field of <code>virtualSpaces</code> in <code>StorageGroups</code> is set to <code>share</code>.
login	Login object	Node login mode. Either the key pair or password must be used for login.
rootVolume	Volume object	Information about disks on the node
dataVolumes	Array of Volume objects	<p>Data disk parameters of the node. Currently, you can add the second data disk for your node on the CCE console.</p> <p>This data disk is used by the container runtime and kubelet. Do not uninstall this disk. Otherwise, the node will become unavailable.</p> <p>For DeC nodes, the parameter description is the same as that for <code>rootVolume</code>.</p>

Parameter	Type	Description
storage	Storage object	<p>Disk initialization management parameter. This parameter is complex to configure. For details, see Attaching Disks to a Node.</p> <p>If this parameter retains its default, disks are managed based on the DockerLVMConfigOverride (discarded) parameter in extendParam. This parameter is supported by clusters of version 1.15.11 and later.</p> <p>NOTE</p> <ul style="list-style-type: none"> • If a node specification involves both local disks and EVS disks, do not retain the default value of this parameter to prevent unexpected disk partitions. • If you want to change the value range of a data disk to 20 to 32768, do not use the default value. • If you want to use the shared disk space (with the runtime and Kubernetes partitions cancelled), do not retain the default value of this parameter. For details about the shared disk space, see Data Disk Space Allocation.
publicIP	NodePublicIP object	<p>EIP of a node.</p> <p>NOTE This parameter is not supported when you add a node to a node pool.</p>
nodeNicSpec	NodeNicSpec object	NIC of the node
count	Integer	Number of nodes to be created in a batch. The value must be a positive integer greater than or equal to 1 and less than or equal to the defined limit. This parameter can be left blank when it is used for a node pool.
billingMode	Integer	<p>Node billing mode.</p> <ul style="list-style-type: none"> • 0: pay-per-use • 1: yearly/monthly • 2: (discarded) yearly/monthly billing with auto payment enabled

Parameter	Type	Description
taints	Array of Taint objects	<p>You can add taints to created nodes to configure anti-affinity. A maximum of 20 taints can be added. Each taint contains the following parameters:</p> <ul style="list-style-type: none"> • Key: A key must contain 1 to 63 characters starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain name can be used as the prefix of a key. • Value: A value must start with a letter or digit and can contain a maximum of 63 characters, including letters, digits, hyphens (-), underscores (_), and periods (.). • Effect: Available options are NoSchedule, PreferNoSchedule, and NoExecute. When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query. <p>Example:</p> <pre>"taints": [{ "key": "status", "value": "unavailable", "effect": "NoSchedule" }, { "key": "looks", "value": "bad", "effect": "NoSchedule" }]</pre>

Parameter	Type	Description
k8sTags	Map<String,String>	<p>Defined in key-value pairs. A maximum of 20 key-value pairs are allowed.</p> <ul style="list-style-type: none"> Key: Enter 1 to 63 characters, starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain can be prefixed to a key and contain a maximum of 253 characters. Example DNS subdomain: example.com/my-key Value: The value can be left blank or contain 1 to 63 characters that start with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed in the character string. When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query. <p>Example:</p> <pre>"k8sTags": { "key": "value" }</pre>
ecsGroupId	String	<p>ECS group ID. If this parameter is configured, nodes will be created in the specified ECS group.</p> <p>NOTE This configuration does not take effect when a node pool is created. To ensure that all nodes in a node pool are in the same ECS group, configure nodeManagement in the node pool.</p>
dedicatedHostId	String	<p>ID of the DeH to which the node is scheduled.</p> <p>NOTE This parameter is not supported when you add a node during node pool creation.</p>

Parameter	Type	Description
userTags	Array of UserTag objects	<p>Cloud server tag. The key of a tag must be unique. The maximum number of custom tags supported by CCE depends on the region and cannot exceed 8.</p> <p>When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query.</p> <p>NOTE Use only letters, Unicode characters, digits, special characters(-, _). Max characters: 36</p>
runtime	Runtime object	<p>Container runtime:</p> <ul style="list-style-type: none"> • Clusters of v1.25 or earlier: docker. • Clusters of v1.25 or later: Container runtime varies with the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker. For nodes running other OSs, the default container runtime is containerd.

Parameter	Type	Description
initializedConditions	Array of strings	<p>Custom initialization flag, which is left blank by default.</p> <p>Before CCE nodes are initialized, node.cloudprovider.kubernetes.io/uninitialized is added to the nodes to prevent pods from being scheduled to these nodes. When creating a node, you can configure the initializedConditions parameter to control the taint removal time. By default, the timeout period is not configured.</p> <p>Example:</p> <ol style="list-style-type: none"> 1. Create a node and add parameter "initializedConditions": ["CCEInitial", "CustomedInitial"]. 2. After custom initialization is complete, CCE will call a Kubernetes API (for example, PATCH /v1/nodes/{node_ip}/status) to update the node conditions by adding two labels of types CCEInitial and CustomedInitial and setting the status to True, as shown in the following: <pre data-bbox="815 1055 1430 1205"> status: conditions: - type: CCEInitial status: 'True' - type: CustomedInitial status: 'True' </pre> <ol style="list-style-type: none"> 1. CCE polls status.Conditions of nodes to check whether there are conditions of types CCEInitial and CustomedInitial. If such conditions exist and the status is True, the node initialization is complete and the initialization taint is removed. 2. initializedConditions allows you to configure a timeout period during node creation. For example, "initializedConditions": ["CCEInitial:15m", "CustomedInitial:15m"] indicates that the timeout period is 15 minutes. After the timeout period is reached, the initialization conditions will be automatically ignored and the initialization taint will be removed when CCE polls the node. <ul style="list-style-type: none"> ● Use only letters and digits. Max. characters: 20. ● Max. flags: 2. ● The unit of the timeout period is minute (m).

Parameter	Type	Description
extendParam	NodeExtendParam object	Extended parameters for creating a node.
hostnameConfig	HostnameConfig object	Kubernetes node name configuration parameter, which is supported by clusters of v1.23.6-r0 to v1.25 or clusters of v1.25.2-r0 or later versions.

Table 4-581 Login

Parameter	Type	Description
sshKey	String	Name of the key pair used for login.
userPassword	UserPassword object	Password used for node login.

Table 4-582 UserPassword

Parameter	Type	Description
username	String	Login account. The default value is root .
password	String	<p>If a username and a password are used to create a node, this field is shielded in the response body.</p> <p>A password must meet the following complexity requirements:</p> <ul style="list-style-type: none"> • Contains 8 to 26 characters. • Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (!@\$%^-_[{}];,./?~#*) • Cannot contain the username or the username spelled backwards. <p>The password field must be salted during node creation. For details, see Adding a Salt in the password Field When Creating a Node.</p>

Table 4-583 Volume

Parameter	Type	Description
size	Integer	Disk size, in GB. <ul style="list-style-type: none"> • System disk: 40 to 1024 • Value range for the first data disk: 20 to 32768 (When storage is not specified, this value range becomes 100 to 32768.) • Value range for other data disks: 10 to 32768. (When storage is not specified, this value range becomes 100 to 32768.)
volumetype	String	Disk type. For details about possible values, see the description of the root_volume parameter in the API used to create an ECS. <ul style="list-style-type: none"> • SAS: high I/O SAS disk • SSD: ultra-high I/O SSD disk • SATA: common I/O SATA disk SATA disks have been removed from EVS. You can find them attached only on existing nodes.
extendParam	Map<String,Object>	Extended disk parameters, defined in extendparam in the API used to create an ECS. For details, see Creating an ECS .
cluster_id	String	ID of the storage pool used by the ECS system disk. This field is used only for DeC clusters, which functions as dssPoolID , that is, the ID of the DSS storage pool. To obtain the value, see the ID field in the API used to obtain details of a DSS storage pool .
cluster_type	String	Storage class of the cloud server system disk. The value is always dss . This field is used only for DeC clusters.
hw:passthrough	Boolean	<ul style="list-style-type: none"> • Pay attention to this field if your ECS is SDI-compliant. If the value of this field is true, an SCSI disk will be created. • If the node pool type is ElasticBMS, this field must be set to true. • If a node specification involves local disks and EVS disks at the same time, set the disk initialization parameters. For details, see Attaching Disks to a Node.
metadata	VolumeMetadata object	EVS disk encryption information. This field is mandatory only when you need to encrypt the system disk or data disks of the node to be created.

Table 4-584 VolumeMetadata

Parameter	Type	Description
__system__encrypted	String	Whether the EVS disk is encrypted. The value 0 indicates that the EVS disk is not encrypted, and the value 1 indicates that the EVS disk is encrypted. If this parameter is not specified, EVS disks will not be encrypted by default.
__system__cmkid	String	CMK ID, which indicates encryption in metadata . This field is used with __system__encrypted .

Table 4-585 Storage

Parameter	Type	Description
storageSelectors	Array of StorageSelectors objects	Disk selection. Matched disks are managed according to matchLabels and storageType .
storageGroups	Array of StorageGroups objects	A storage group consists of multiple storage devices. It is used to divide storage space.

Table 4-586 StorageSelectors

Parameter	Type	Description
name	String	Selector name, used as the index of selectorNames in storageGroup . Therefore, the name of each selector must be unique.
storageType	String	Specifies the storage type. Currently, only evs (EVS volumes) and local (local volumes) are supported. The local storage does not support disk selection. All local disks will form a VG. Therefore, only one storageSelector of the local type is allowed.
matchLabels	matchLabels object	Matching field of an EVS volume. The size , volumeType , metadataEncrypted , metadataCmkid and count fields are supported.

Table 4-587 matchLabels

Parameter	Type	Description
size	String	Matched disk size. If this parameter is left unspecified, the disk size is not limited. Example: 100
volumeType	String	EVS disk type. SSD, GPSSD, SAS, ESSD, and SATA are supported. If this parameter is left blank, there is no restriction on the disk type.
metadataEncrypted	String	Disk encryption identifier. If this parameter is set to 0 , it specifies that the disk is not encrypted. If this parameter is set to 1 , it specifies that the disk is encrypted. If this parameter is left blank, there is no restriction on the disk encryption identifier.
metadataCmkid	String	CMK ID of an encrypted disk. The value is a string of 36 bytes. If this parameter is left blank, there is no restriction on the disk key ID.
count	String	Number of disks to be selected. If this parameter is left blank, all disks of this type are selected.

Table 4-588 StorageGroups

Parameter	Type	Description
name	String	Name of a virtual storage group, which must be unique. NOTE <ul style="list-style-type: none"> If cceManaged is set to true, the name must be set to vgpaas. If the data disk is used as a temporary storage volume, the name must be vg-everest-localvolume-ephemeral. If the data disk is used as a persistent storage volume, the name must be vg-everest-localvolume-persistent.
cceManaged	Boolean	Storage space for Kubernetes and runtime components. Only one group can be set to true . If this parameter is left blank, the default value false is used.
selectorNames	Array of strings	This parameter corresponds to name in storageSelectors . A group can match multiple selectors, but a selector can match only one group.

Parameter	Type	Description
virtualSpaces	Array of VirtualSpace objects	Detailed management of space configuration in a group.

Table 4-589 VirtualSpace

Parameter	Type	Description
name	String	Name of a virtualSpace. Options: share , kubernetes , runtime , and user <ul style="list-style-type: none"> • share: shared space configuration (with the runtime and Kubernetes partitions cancelled). lvmConfig needs to be configured. • Kubernetes: Kubernetes space configuration. lvmConfig needs to be configured. • runtime: runtime space configuration. runtimeConfig needs to be configured. • user: user space configuration. lvmConfig needs to be configured.
size	String	Size of a virtualSpace. The value must be an integer in percentage. Example: 90%. NOTE The sum of the percentages of all virtualSpaces in a group cannot exceed 100%.
lvmConfig	LVMConfig object	LVM configuration managements, applicable to the <i>share</i> *, kubernetes , and user spaces. Note that one virtual space supports only one config.
runtimeConfig	RuntimeConfig object	runtime configurations, applicable to the runtime space. Note that one virtual space supports only one config.

Table 4-590 LVMConfig

Parameter	Type	Description
lvType	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.

Parameter	Type	Description
path	String	Path to which the disk is attached. This parameter takes effect only in user configuration. The value is an absolute path. Digits, letters, periods (.), hyphens (-), and underscores (_) are allowed.

Table 4-591 RuntimeConfig

Parameter	Type	Description
lvType	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.

Table 4-592 NodePublicIP

Parameter	Type	Description
ids	Array of strings	IDs of existing EIPs. The quantity cannot be greater than the number of nodes to be created. NOTE If ids has been set, you do not need to set count and eip .
count	Integer	Number of EIPs to be dynamically created. NOTE count and eip must be set at the same time.
eip	NodeEIPSpec object	EIP configuration.

Table 4-593 NodeEIPSpec

Parameter	Type	Description
iptype	String	EIP type, specified in publicip.type in the API for assigning an EIP. For details, see Assigning an EIP .
bandwidth	NodeBandwidth object	Bandwidth parameters of the EIP

Table 4-594 NodeBandwidth

Parameter	Type	Description
chargemode	String	<p>Bandwidth billing mode.</p> <ul style="list-style-type: none"> • If this field is not specified, the billing is based on bandwidth. • If the field is null, the billing is based on bandwidth. • If the field value is traffic, the billing is based on traffic. • If the value is out of the preceding options, the cloud server will fail to be created. <p>NOTE</p> <ul style="list-style-type: none"> • Billed by bandwidth: The billing will be based on the data transmission rate (in Mbps) of public networks. This billing mode is recommended if your bandwidth usage is higher than 10%. • Billed by traffic: The billing is based on the total amount of data (in GB) transmitted over the public network. This mode is available only when you are creating a pay-per-use node. This billing mode is recommended if your bandwidth usage is lower than 10%.
size	Integer	<p>Bandwidth size, specified in bandwidth.size in the API for assigning an EIP.</p> <p>For details, see Assigning an EIP.</p>
sharetype	String	<p>Bandwidth sharing type. Value options: PER (exclusive bandwidth)</p>

Table 4-595 NodeNicSpec

Parameter	Type	Description
primaryNic	NicSpec object	Description of the primary NIC.
extNics	Array of NicSpec objects	<p>Extension NIC</p> <p>NOTE This parameter is not supported when you add a node to a node pool.</p>

Table 4-596 NicSpec

Parameter	Type	Description
subnetId	String	Network ID of the subnet to which a NIC belongs. If subnetId is not specified when a primary NIC is creating, the cluster subnet will be used. If subnetList is also configured for a node pool, the subnetList field is used for adding subnets to the node pool. When creating an extension NIC, you must specify subnetId .
fixedIps	Array of strings	The IP address of the primary ENI is specified using fixedIps . The number of IP addresses cannot be greater than the number of created nodes. Either fixedIps or ipBlock can be specified. fixedIps cannot be specified for extension ENIs.
ipBlock	String	CIDR format of the primary NIC IP range. The IP address of the created node falls in this range. fixedIps and ipBlock cannot be specified at the same time.

Table 4-597 Taint

Parameter	Type	Description
key	String	Key.
value	String	Value.
effect	String	Effect.

Table 4-598 UserTag

Parameter	Type	Description
key	String	Key of the cloud server label. The value cannot start with CCE- or __type_baremetal .
value	String	Value of the cloud server label.

Table 4-599 Runtime

Parameter	Type	Description
name	String	<p>Container runtime. Default value:</p> <ul style="list-style-type: none"> • Clusters earlier than v1.25: The default value is docker. • Clusters of v1.25 or later: The default container runtime varies depending on the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker; for nodes running other OSs, the default container runtime is containerd.

Table 4-600 NodeExtendParam

Parameter	Type	Description
ecs:performancetype	String	ECS flavor types. This field is returned in the response.
orderId	String	Order ID. This parameter is returned in the response when the node is billed on a yearly/monthly basis with auto payment enabled (only in creation scenarios).
productId	String	Product ID. This field is returned in the response when the node is yearly/monthly-billed with auto payment enabled.
maxPods	Integer	<p>Maximum number of pods that can be created on a node, including the default system pods. Value range: 16 to 256.</p> <p>This limit prevents the node from being overloaded of pods.</p> <p>The number of pods that can be created on a node is determined by multiple parameters. For details, see Maximum Number of Pods That Can Be Created on a Node.</p>
periodType	String	<ul style="list-style-type: none"> • month: The unit is month. • year: The unit is year. <p>NOTE</p> <p>Request parameter, which is valid and mandatory only when billingMode is set to 1 (yearly/monthly billing) or 2 (yearly/monthly billing with auto renewal enabled) (discarded).</p> <p>Response parameter, which is returned only when a yearly/monthly node is created.</p>

Parameter	Type	Description
periodNum	Integer	<p>Subscription duration. The value can be:</p> <ul style="list-style-type: none"> • If periodType is month, the value ranges from 1 to 9. • If periodType is year, the value is 1. <p>NOTE Request parameter, which is valid and mandatory when billingMode is set to 1 or 2 (discarded). Response parameter, which is returned only when a yearly/monthly node is created.</p>
isAutoRenew	String	<p>Whether auto renewal is enabled.</p> <ul style="list-style-type: none"> • true: Auto renewal is enabled. • false: Auto renewal is not enabled. <p>NOTE This field is valid when billingMode is set to 1 or 2 (discarded). If not specified, auto renewal is not enabled.</p>
isAutoPay	String	<p>Whether to deduct fees automatically.</p> <ul style="list-style-type: none"> • true: Enable automatic fee deduction. • false: Do not enable automatic fee deduction. <p>NOTE This field is valid when billingMode is set to 1 or 2 (discarded). If not specified when billingMode is set to 1, auto fee deduction is not enabled. (Discarded: If not specified when billingMode is set to 2, auto fee deduction is enabled.)</p>

Parameter	Type	Description
DockerLVMConfigOverride	String	<p>Docker data disk configuration item. (This parameter has been discarded. Use the storage field instead.) Example default configuration: "DockerLVMConfigOverride":"dockerThinpool=vgpaas/90%VG;kubernetesLV=vgpaas/10%VG;diskType=evs;lvType=linear"</p> <p>By default, if no VD disk is available, an error occurs because the data disk fails to be found. Set diskType based on the actual drive letter type.</p> <p>The following fields are included:</p> <ul style="list-style-type: none"> • userLV (optional): size of the user space, for example, vgpaas/20%VG. • userPath (optional): mount path of the user space, for example, /home/wqt-test. • diskType: disk type. Currently, only evs, hdd, and ssd are supported. • lvType: type of a logic volume. The value can be linear or striped. • dockerThinpool: Docker space size, for example, vgpaas/60%VG. • kubernetesLV: kubelet space size, for example, vgpaas/20%VG.
dockerBaseSize	Integer	<p>Available disk space of a single container on a node, in GB. This parameter has been discarded. Use containerBaseSize.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In Device Mapper mode, the default value is 10. In OverlayFS mode, the available space of a single container is not limited by default, and the dockerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters of the new versions.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that dockerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device.</p>

Parameter	Type	Description
containerBaseSize	Integer	<p>Available disk space of a single container on a node, in GB.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In OverlayFS mode, the available space of a single container is not limited by default. In Device Mapper mode, the default value is 10, and the containerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters v1.23.14-r0, v1.25.9-r0, v1.27.6-r0, v1.28.4-r0 or later.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that containerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device. In new versions, the Device Mapper mode is used only by BMSs in the same resource pool and is being discarded.</p>
publicKey	String	Public key of a node.
alpha.cce/preInstall	String	<p>Pre-installation script.</p> <p>NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)</p>
alpha.cce/postInstall	String	<p>Post-installation script.</p> <p>NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)</p>
alpha.cce/NodeImageID	String	This parameter is required when a custom image is used to create a BMS node.

Parameter	Type	Description
nicMultiqueue	String	<ul style="list-style-type: none"> Number of ENI queues. Example setting: <code>"[{"queue":4}]"</code> The following fields are included: <ul style="list-style-type: none"> queue: number of ENI queues. This field can be configured only for BMS nodes in a CCE Turbo cluster. Supported proportions are {"1":128, "2":92, "4":92, "8":32, "16":16, "28":9}. That is, if there is one queue, a maximum of 128 ENIs can be bound. If there are two queues, a maximum of 92 ENIs can be bound for two queues. A larger number of ENI queues indicates higher performance but fewer ENIs can be bound. The queue settings cannot be changed after the node pool is created.
nicThreshold	String	<ul style="list-style-type: none"> ENI pre-binding thresholds. Example setting: <code>"0.3:0.6"</code> <ul style="list-style-type: none"> Low threshold (L): determines the minimum number of pre-bound ENIs (Min). Formula: $Min = Total\ number\ of\ ENIs\ of\ the\ node \times L$ High threshold (H): determines the maximum number of pre-bound ENIs (Max). Formula: $Max = Total\ number\ of\ ENIs\ of\ the\ node \times H$ Number of ENIs bound to a BMS node (B) and number of ENIs being used by pods (U): $U + Min < B < U + Max$ If the number of pre-bound ENIs on a BMS node is smaller than the minimum allowed, the system will bind more ENIs to make the numbers equal. If the number of pre-bound ENIs on a BMS node is larger than the maximum allowed, the system periodically unbinds ENIs (about every 2 minutes) to make the numbers equal. Both the thresholds are one-decimal-place values ranging from 0.0 to 1.0. The low threshold must be smaller than or equal to the high one. This field can be configured only for BMS nodes in a CCE Turbo cluster. Pre-binding ENIs can speed up workload creation but occupies IP addresses.

Parameter	Type	Description
chargingMode	Integer	Billing mode of a node. This parameter has been deprecated. Use the billingMode parameter in NodeSpec .
agency_name	String	Name of an agency An agency is created by a tenant administrator on Identity and Access Management (IAM) to provide temporary credentials for CCE nodes to access cloud servers. This parameter is returned only when it is transferred during node creation.
kubeReservedMem	Integer	Reserved node memory, which is reserved for Kubernetes components.
systemReservedMem	Integer	Reserved node memory, which is reserved for system components.
init-node-password	String	Node password. If this parameter is used as a response parameter, asterisks (*) will be displayed.

Table 4-601 HostnameConfig

Parameter	Type	Description
type	String	Configuration type of the Kubernetes node name. The default value is privatelp . <ul style="list-style-type: none"> • privatelp: The Kubernetes node is named after its IP address. • cceNodeName: The Kubernetes node is named after the CCE node. <p>NOTE</p> <ul style="list-style-type: none"> • For a node which is configured using cceNodeName, the name is the same as the Kubernetes node name and the ECS name. The node name cannot be changed. If the ECS name is changed on the ECS console, the node name will retain unchanged after ECS synchronization. • For a node which is configured using cceNodeName, to avoid a conflict between Kubernetes nodes, the system automatically adds a suffix to each node name. The suffix is in the format of A hyphen (-) Five random characters. The value of the random characters is a lowercase letter or a digit ranging from 0 to 9.

Table 4-602 NodePoolNodeAutoscaling

Parameter	Type	Description
enable	Boolean	Whether to enable auto scaling.
minNodeCount	Integer	Minimum number of nodes allowed if auto scaling is enabled. The value cannot be greater than the maximum number of nodes allowed by the cluster specifications.
maxNodeCount	Integer	Maximum number of nodes allowed if auto scaling is enabled. This value must be greater than or equal to the value of minNodeCount and cannot exceed the maximum number of nodes in the cluster specifications.
scaleDownCooldownTime	Integer	Interval between two scaling operations, in minutes. During this period, nodes added after a scale-up will not be deleted.
priority	Integer	Weight of a node pool. A node pool with a higher weight has a higher priority during scaling.

Table 4-603 NodeManagement

Parameter	Type	Description
serverGroupReference	String	Cloud server group ID. If this field is specified, all nodes in the node pool will be created in this group. The group ID can be specified only when you create the node pool and cannot be modified. When you specify a cloud server group, the number of nodes in the node pool cannot exceed the group quota.

Table 4-604 SecurityID

Parameter	Type	Description
id	String	Security group ID

Table 4-605 NodePoolStatus

Parameter	Type	Description
currentNode	Integer	Total number of nodes in the current node pool (excluding the nodes that are being deleted)

Parameter	Type	Description
creatingNode	Integer	Number of nodes in the creation process in the node pool
deletingNode	Integer	Number of nodes being deleted in the current node pool.
phase	String	<p>Node pool status.</p> <ul style="list-style-type: none"> • Null: available (the number of current nodes in the node pool has reached the expected value, and no node scaling is being performed.) • Synchronizing: scaling in progress (the number of current nodes in the node pool does not reach the expected value and no node scaling is being performed.) • Synchronized: pending scaling (the number of current nodes in the node pool does not reach the expected value, or node scaling is being performed.) • SoldOut: The node pool cannot be scaled out. (This field is used in multiple scenarios, for example, resources in the node pool have been sold out or the resource quota is insufficient.) <p>NOTE This node pool status has been discarded and is reserved only for compatibility. You are not advised to use it. The replacements are as follows:</p> <ul style="list-style-type: none"> • Node pool scaling status: You can obtain accurate status of the current node pool based on parameters such as currentNode, creatingNode, or deletingNode. • Node pool scale-out: You can use conditions to obtain the detailed status of a node pool. Scalable can replace SoldOut. • Deleting: The object is being deleted. • Error: An error occurs.
jobId	String	ID of a job executed on the node pool. This field is available only when the node pool is being deleted.
conditions	Array of NodePoolCondition objects	Node pool status details. For details, see the definition of Condition .

Table 4-606 NodePoolCondition

Parameter	Type	Description
type	String	Condition type. The options are as follows: <ul style="list-style-type: none"> • Scalable: whether a node pool can be scaled. If the status is False, node pool scaling will not be triggered again. • QuotaInsufficient: Quotas on which node pool scaling depends are insufficient, affecting the node pool scaling status. • ResourceInsufficient: Resources on which node pool scaling depends are insufficient, affecting the node pool scaling status. • UnexpectedError: The node pool fails to be scaled out due to unexpected reasons, affecting the node pool scaling status. • LockedByOrder: The yearly/monthly-billed node pool is locked due to an order. The value of Reason is the ID of the pending order. • Error: A node pool error occurs. A common trigger is deletion failure.
status	String	Current status of Condition . The options are as follows: <ul style="list-style-type: none"> • "True" • "False"
lastProbeTime	String	Time when the status was last checked
lastTransitTime	String	Time when the status was last changed
reason	String	Reason why the status was last changed
message	String	Detailed condition description

Example Requests

None

Example Responses

Status code: 200

Information about all node pools in the cluster is successfully obtained.

```
{
  "kind": "List",
  "apiVersion": "v3",
  "items": [{
    "kind": "NodePool",
```

```
"apiVersion" : "v3",
"metadata" : {
  "name" : "az1.dc1#s1.large#EulerOS 2.2",
  "uid" : "az1.dc1#s1.large#EulerOS 2.2"
},
"spec" : {
  "nodeTemplate" : {
    "flavor" : "s1.large",
    "az" : "az1.dc1",
    "os" : "EulerOS 2.2",
    "login" : {
      "sshKey" : "KeyPair-001"
    },
    "rootVolume" : { },
    "publicIP" : {
      "eip" : {
        "bandwidth" : { }
      }
    }
  },
  "billingMode" : 0
},
"autoscaling" : {
  "enable" : true,
  "maxNodeCount" : 50
}
},
"status" : {
  "currentNode" : 1
}
}
}]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ListNodePoolsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
```

```
        .withCredential(auth)
        .withRegion(CceRegion.valueOf("<YOUR REGION>"))
        .build();
ListNodePoolsRequest request = new ListNodePoolsRequest();
request.withClusterId("{cluster_id}");
try {
    ListNodePoolsResponse response = client.listNodePools(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListNodePoolsRequest()
        request.cluster_id = "{cluster_id}"
        response = client.list_node_pools(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
```

```

)
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListNodePoolsRequest{}
    request.ClusterId = "{cluster_id}"
    response, err := client.ListNodePools(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Information about all node pools in the cluster is successfully obtained.

Error Codes

See [Error Codes](#).

4.4.4 Updating a Specified Node Pool

Function

This API is used to update information about a specified node pool. This API can be called only when the cluster is in the Available, Scale Out, or Scale In state.

 NOTE

- The URL for cluster management is in the following format: https://Endpoint/uri. uri indicates the resource path, that is, the API access path.
- Currently, only the node pool name can be updated. initialNodeCount, k8sTags, taints, login and userTags under spec are related to the scaling configuration of the node pool. If no value is set for this update, the initial value is used by default.

Calling Method

For details, see [Calling APIs](#).

URI

PUT /api/v3/projects/{project_id}/clusters/{cluster_id}/nodepools/{nodepool_id}

Table 4-607 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
nodepool_id	Yes	String	Node pool ID.

Request Parameters

Table 4-608 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Table 4-609 Request body parameters

Parameter	Mandatory	Type	Description
metadata	Yes	NodePoolMetadataUpdate object	Metadata information of the node pool.
spec	Yes	NodePoolSpecUpdate object	Node pool specifications.

Table 4-610 NodePoolMetadataUpdate

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the node pool. NOTE Naming rules: <ul style="list-style-type: none"> Enter 1 to 50 characters, starting with a lowercase letter and not ending with a hyphen (-). Only lowercase letters, digits, and hyphens (-) are allowed. You cannot create node pools named DefaultPool.

Table 4-611 NodePoolSpecUpdate

Parameter	Mandatory	Type	Description
nodeTemplate	Yes	NodeSpecUpdate object	Detailed parameters of the node pool template.
initialNodeCount	Yes	Integer	Initial number of nodes for the node pool. When queried, the value is the number of target nodes in the node pool. The default value is 0 .
autoscaling	Yes	NodePoolNodeAutoscaling object	Auto scaling parameters. Only pay-per-use node pools support auto scaling.

Table 4-612 NodeSpecUpdate

Parameter	Mandatory	Type	Description
taints	Yes	Array of Taint objects	<p>You can add taints to created nodes to configure anti-affinity. A maximum of 20 taints can be added. This parameter is left blank by default. Each taint contains the following parameters:</p> <ul style="list-style-type: none"> • Key: Enter 1 to 63 characters starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain can be prefixed to a key. • Value: Enter 1 to 63 characters with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. • Effect: Available options are NoSchedule, PreferNoSchedule, and NoExecute. <p>Example:</p> <pre>"taints": [{ "key": "status", "value": "unavailable", "effect": "NoSchedule" }, { "key": "looks", "value": "bad", "effect": "NoSchedule" }]</pre> <p>NOTE If this parameter is not specified or left empty, the custom taints of the node pool will be deleted.</p>

Parameter	Mandatory	Type	Description
k8sTags	Yes	Map<String,String>	<p>Defined in key-value pairs. A maximum of 20 key-value pairs are allowed. This parameter is left blank by default.</p> <ul style="list-style-type: none"> • Key: Enter 1 to 63 characters starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain containing a maximum of 253 characters (such as example.com/my-key) can be prefixed to a key. • Value: The value can be left blank or contain 1 to 63 characters that start with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. <p>Example:</p> <pre>"k8sTags": { "key": "value" }</pre> <p>NOTE If this parameter is not specified or left empty, the custom Kubernetes labels of the node pool will be deleted.</p>
userTags	Yes	Array of UserTag objects	<p>Cloud server tag. The key of a tag must be unique. The maximum number of custom tags supported by CCE varies depending on regions and cannot exceed 8. This parameter is left blank by default.</p> <p>NOTE If this parameter is not specified or left empty, the custom cloud server tags of the node pool will be deleted.</p>

Parameter	Mandatory	Type	Description
initializedConditions	No	Array of strings	<p>Custom initialization flag, which is left blank by default. Before CCE nodes are initialized, node.cloudprovider.kubernetes.io/uninitialized is added to the nodes to prevent pods from being scheduled to these nodes. When creating a node, you can configure the initializedConditions parameter to control the taint removal time. By default, the timeout period is not configured.</p> <p>Example:</p> <ol style="list-style-type: none"> 1. Create a node and add parameter <code>"initializedConditions": ["CCEInitial", "CustomedInitial"]</code>. 2. After custom initialization is complete, CCE will call a Kubernetes API (for example, PATCH /v1/nodes/{node_ip}/status) to update the node conditions by adding two labels of types CCEInitial and CustomedInitial and setting the status to True, as shown in the following: <pre>status: conditions: - type: CCEInitial status: 'True' - type: CustomedInitial status: 'True'</pre> <ol style="list-style-type: none"> 1. CCE polls status.Conditions of nodes to check whether there are conditions of types CCEInitial and CustomedInitial. If such conditions exist and the status is True, the node initialization is complete and the initialization taint is removed.

Parameter	Mandatory	Type	Description
			<p>2. initializedConditions allows you to configure a timeout period during node creation. For example, "initializedConditions": ["CCEInitial:15m", "CustomedInitial:15m"] indicates that the timeout period is 15 minutes. After the timeout period is reached, the initialization conditions will be automatically ignored and the initialization taint will be removed when CCE polls the node.</p> <ul style="list-style-type: none"> • Use only letters and digits. Max. characters: 20. • Max. flags: 2. • The unit of the timeout period is minute (m).
login	No	Login object	Node login mode. Either the key pair login mode or the password login mode must be used.

Table 4-613 Taint

Parameter	Mandatory	Type	Description
key	Yes	String	Key.
value	No	String	Value.
effect	Yes	String	Effect.

Table 4-614 UserTag

Parameter	Mandatory	Type	Description
key	No	String	Key of the cloud server label. The value cannot start with CCE- or __type_baremetal .
value	No	String	Value of the cloud server label.

Table 4-615 Login

Parameter	Mandatory	Type	Description
sshKey	No	String	Name of the key pair used for login.
userPassword	No	UserPassword object	Password used for node login.

Table 4-616 UserPassword

Parameter	Mandatory	Type	Description
username	No	String	Login account. The default value is root .
password	Yes	String	<p>If a username and a password are used to create a node, this field is shielded in the response body.</p> <p>A password must meet the following complexity requirements:</p> <ul style="list-style-type: none"> • Contains 8 to 26 characters. • Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (!@\$%^-_=+[{ }];,./?~#*) • Cannot contain the username or the username spelled backwards. <p>The password field must be salted during node creation. For details, see Adding a Salt in the password Field When Creating a Node.</p>

Table 4-617 NodePoolNodeAutoscaling

Parameter	Mandatory	Type	Description
enable	No	Boolean	Whether to enable auto scaling.

Parameter	Mandatory	Type	Description
minNodeCount	No	Integer	Minimum number of nodes allowed if auto scaling is enabled. The value cannot be greater than the maximum number of nodes allowed by the cluster specifications.
maxNodeCount	No	Integer	Maximum number of nodes allowed if auto scaling is enabled. This value must be greater than or equal to the value of minNodeCount and cannot exceed the maximum number of nodes in the cluster specifications.
scaleDownCooldownTime	No	Integer	Interval between two scaling operations, in minutes. During this period, nodes added after a scale-up will not be deleted.
priority	No	Integer	Weight of a node pool. A node pool with a higher weight has a higher priority during scaling.

Response Parameters

Status code: 200

Table 4-618 Response body parameters

Parameter	Type	Description
kind	String	API type. The value is fixed at NodePool .
apiVersion	String	API version. The value is fixed at v3 .
metadata	NodePoolMetadata object	Metadata of the node pool.
spec	NodePoolSpec object	Node pool specifications.
status	UpdateNodePoolStatus object	Node pool status.

Table 4-619 NodePoolMetadata

Parameter	Type	Description
name	String	Node pool name. NOTE Naming rules: Enter 1 to 50 characters, starting with a lowercase letter and not ending with a hyphen (-). Only lowercase letters, digits, and hyphens (-) are allowed. <ul style="list-style-type: none"> You cannot create node pools named DefaultPool.
uid	String	UID of the node pool. The value is automatically generated after the object is updated. A user-defined value will not take effect.
annotations	Map<String,String>	Annotations of a node pool in key-value pairs. This parameter is used only for query and cannot be input through a request. Entered data of this parameter is invalid.
updateTimestamp	String	Update time.
creationTimestamp	String	Creation time.

Table 4-620 NodePoolSpec

Parameter	Type	Description
type	String	Node pool type. If this parameter is left blank, the value vm is used by default. <ul style="list-style-type: none"> vm: ECS ElasticBMS: C6 general computing-plus BMS. An example flavor is c6.22xlarge.2.physical. bms: BMS
nodeTemplate	NodeSpec object	Detailed parameters of the node pool template.
initialNodeCount	Integer	Initial number of nodes for the node pool. When queried, the value is the number of target nodes in the node pool.
autoscaling	NodePoolNodeAutoscaling object	Auto scaling parameters

Parameter	Type	Description
nodeManagement	NodeManagement object	Node management configuration
podSecurityGroups	Array of SecurityID objects	Security group configuration. This parameter is available only for CCE Turbo clusters.
customSecurityGroups	Array of strings	<p>Custom security group settings for a node pool. New nodes scaled out in a node pool can be bound to a specified security group.</p> <ul style="list-style-type: none"> • Specifying no security group ID will add the new nodes to the default security group of the worker nodes. • Specifying a valid security group ID will put new nodes in that security group. • When specifying a security group, do not modify the rules of the port on which CCE running depends. For details, see How Do I Harden the Automatically Created Security Group Rules for CCE Cluster Nodes.

Table 4-621 NodeSpec

Parameter	Type	Description
flavor	String	Node specifications. For details about the node specifications supported by CCE, see Node Flavor Description .
az	String	<p>AZ where the node to be created is located. You need to specify the AZ name. If it is set to random, the node will be created in a random AZ.</p> <p>For details about AZs supported by CCE, see Regions and Endpoints.</p>

Parameter	Type	Description
os	String	<p>Node OS. For details about the supported OSs, see Node OS.</p> <p>NOTE</p> <ul style="list-style-type: none"> The system automatically selects the supported OS based on the cluster version. If the current cluster version does not support the OS, an error will be reported. If <code>alpha.cce/NodeImageID</code> in <code>extendParam</code> is specified during node creation, you do not need to configure this parameter. This parameter is mandatory when creating a node pool. If you are using shared disk space when creating a node, this parameter is mandatory. Specifically, the disk initialization configuration management parameter <code>storage</code> is used and the <code>name</code> field of <code>virtualSpaces</code> in <code>StorageGroups</code> is set to <code>share</code>.
login	Login object	Node login mode. Either the key pair or password must be used for login.
rootVolume	Volume object	Information about disks on the node
dataVolumes	Array of Volume objects	<p>Data disk parameters of the node. Currently, you can add the second data disk for your node on the CCE console.</p> <p>This data disk is used by the container runtime and kubelet. Do not uninstall this disk. Otherwise, the node will become unavailable.</p> <p>For DeC nodes, the parameter description is the same as that for <code>rootVolume</code>.</p>

Parameter	Type	Description
storage	Storage object	<p>Disk initialization management parameter. This parameter is complex to configure. For details, see Attaching Disks to a Node.</p> <p>If this parameter retains its default, disks are managed based on the DockerLVMConfigOverride (discarded) parameter in extendParam. This parameter is supported by clusters of version 1.15.11 and later.</p> <p>NOTE</p> <ul style="list-style-type: none"> • If a node specification involves both local disks and EVS disks, do not retain the default value of this parameter to prevent unexpected disk partitions. • If you want to change the value range of a data disk to 20 to 32768, do not use the default value. • If you want to use the shared disk space (with the runtime and Kubernetes partitions cancelled), do not retain the default value of this parameter. For details about the shared disk space, see Data Disk Space Allocation.
publicIP	NodePublicIP object	<p>EIP of a node.</p> <p>NOTE This parameter is not supported when you add a node to a node pool.</p>
nodeNicSpec	NodeNicSpec object	NIC of the node
count	Integer	Number of nodes to be created in a batch. The value must be a positive integer greater than or equal to 1 and less than or equal to the defined limit. This parameter can be left blank when it is used for a node pool.
billingMode	Integer	<p>Node billing mode.</p> <ul style="list-style-type: none"> • 0: pay-per-use • 1: yearly/monthly • 2: (discarded) yearly/monthly billing with auto payment enabled

Parameter	Type	Description
taints	Array of Taint objects	<p>You can add taints to created nodes to configure anti-affinity. A maximum of 20 taints can be added. Each taint contains the following parameters:</p> <ul style="list-style-type: none"> • Key: A key must contain 1 to 63 characters starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain name can be used as the prefix of a key. • Value: A value must start with a letter or digit and can contain a maximum of 63 characters, including letters, digits, hyphens (-), underscores (_), and periods (.). • Effect: Available options are NoSchedule, PreferNoSchedule, and NoExecute. When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query. <p>Example:</p> <pre>"taints": [{ "key": "status", "value": "unavailable", "effect": "NoSchedule" }, { "key": "looks", "value": "bad", "effect": "NoSchedule" }]</pre>

Parameter	Type	Description
k8sTags	Map<String,String>	<p>Defined in key-value pairs. A maximum of 20 key-value pairs are allowed.</p> <ul style="list-style-type: none"> Key: Enter 1 to 63 characters, starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain can be prefixed to a key and contain a maximum of 253 characters. Example DNS subdomain: example.com/my-key Value: The value can be left blank or contain 1 to 63 characters that start with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed in the character string. When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query. <p>Example:</p> <pre>"k8sTags": { "key": "value" }</pre>
ecsGroupId	String	<p>ECS group ID. If this parameter is configured, nodes will be created in the specified ECS group.</p> <p>NOTE This configuration does not take effect when a node pool is created. To ensure that all nodes in a node pool are in the same ECS group, configure nodeManagement in the node pool.</p>
dedicatedHostId	String	<p>ID of the DeH to which the node is scheduled.</p> <p>NOTE This parameter is not supported when you add a node during node pool creation.</p>

Parameter	Type	Description
userTags	Array of UserTag objects	<p>Cloud server tag. The key of a tag must be unique. The maximum number of custom tags supported by CCE depends on the region and cannot exceed 8.</p> <p>When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query.</p> <p>NOTE Use only letters, Unicode characters, digits, special characters(-, _). Max characters: 36</p>
runtime	Runtime object	<p>Container runtime:</p> <ul style="list-style-type: none"> • Clusters of v1.25 or earlier: docker. • Clusters of v1.25 or later: Container runtime varies with the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker. For nodes running other OSs, the default container runtime is containerd.

Parameter	Type	Description
initializedConditions	Array of strings	<p>Custom initialization flag, which is left blank by default.</p> <p>Before CCE nodes are initialized, node.cloudprovider.kubernetes.io/uninitialized is added to the nodes to prevent pods from being scheduled to these nodes. When creating a node, you can configure the initializedConditions parameter to control the taint removal time. By default, the timeout period is not configured.</p> <p>Example:</p> <ol style="list-style-type: none"> 1. Create a node and add parameter "initializedConditions": ["CCEInitial", "CustomedInitial"]. 2. After custom initialization is complete, CCE will call a Kubernetes API (for example, PATCH /v1/nodes/{node_ip}/status) to update the node conditions by adding two labels of types CCEInitial and CustomedInitial and setting the status to True, as shown in the following: <pre>status: conditions: - type: CCEInitial status: 'True' - type: CustomedInitial status: 'True'</pre> <ol style="list-style-type: none"> 1. CCE polls status.Conditions of nodes to check whether there are conditions of types CCEInitial and CustomedInitial. If such conditions exist and the status is True, the node initialization is complete and the initialization taint is removed. 2. initializedConditions allows you to configure a timeout period during node creation. For example, "initializedConditions": ["CCEInitial:15m", "CustomedInitial:15m"] indicates that the timeout period is 15 minutes. After the timeout period is reached, the initialization conditions will be automatically ignored and the initialization taint will be removed when CCE polls the node. <ul style="list-style-type: none"> ● Use only letters and digits. Max. characters: 20. ● Max. flags: 2. ● The unit of the timeout period is minute (m).

Parameter	Type	Description
extendParam	NodeExtendParam object	Extended parameters for creating a node.
hostnameConfig	HostnameConfig object	Kubernetes node name configuration parameter, which is supported by clusters of v1.23.6-r0 to v1.25 or clusters of v1.25.2-r0 or later versions.

Table 4-622 Login

Parameter	Type	Description
sshKey	String	Name of the key pair used for login.
userPassword	UserPassword object	Password used for node login.

Table 4-623 UserPassword

Parameter	Type	Description
username	String	Login account. The default value is root .
password	String	<p>If a username and a password are used to create a node, this field is shielded in the response body.</p> <p>A password must meet the following complexity requirements:</p> <ul style="list-style-type: none"> • Contains 8 to 26 characters. • Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (!@\$%^-_[{}];,./?~#*) • Cannot contain the username or the username spelled backwards. <p>The password field must be salted during node creation. For details, see Adding a Salt in the password Field When Creating a Node.</p>

Table 4-624 Volume

Parameter	Type	Description
size	Integer	Disk size, in GB. <ul style="list-style-type: none"> System disk: 40 to 1024 Value range for the first data disk: 20 to 32768 (When storage is not specified, this value range becomes 100 to 32768.) Value range for other data disks: 10 to 32768. (When storage is not specified, this value range becomes 100 to 32768.)
volumetype	String	Disk type. For details about possible values, see the description of the root_volume parameter in the API used to create an ECS. <ul style="list-style-type: none"> SAS: high I/O SAS disk SSD: ultra-high I/O SSD disk SATA: common I/O SATA disk SATA disks have been removed from EVS. You can find them attached only on existing nodes.
extendParam	Map<String,Object>	Extended disk parameters, defined in extendparam in the API used to create an ECS. For details, see Creating an ECS .
cluster_id	String	ID of the storage pool used by the ECS system disk. This field is used only for DeC clusters, which functions as dssPoolID , that is, the ID of the DSS storage pool. To obtain the value, see the ID field in the API used to obtain details of a DSS storage pool .
cluster_type	String	Storage class of the cloud server system disk. The value is always dss . This field is used only for DeC clusters.
hw:passthrough	Boolean	<ul style="list-style-type: none"> Pay attention to this field if your ECS is SDI-compliant. If the value of this field is true, an SCSI disk will be created. If the node pool type is ElasticBMS, this field must be set to true. If a node specification involves local disks and EVS disks at the same time, set the disk initialization parameters. For details, see Attaching Disks to a Node.
metadata	VolumeMetadata object	EVS disk encryption information. This field is mandatory only when you need to encrypt the system disk or data disks of the node to be created.

Table 4-625 VolumeMetadata

Parameter	Type	Description
__system__encrypted	String	Whether the EVS disk is encrypted. The value 0 indicates that the EVS disk is not encrypted, and the value 1 indicates that the EVS disk is encrypted. If this parameter is not specified, EVS disks will not be encrypted by default.
__system__cmkid	String	CMK ID, which indicates encryption in metadata . This field is used with __system__encrypted .

Table 4-626 Storage

Parameter	Type	Description
storageSelectors	Array of StorageSelectors objects	Disk selection. Matched disks are managed according to matchLabels and storageType .
storageGroups	Array of StorageGroups objects	A storage group consists of multiple storage devices. It is used to divide storage space.

Table 4-627 StorageSelectors

Parameter	Type	Description
name	String	Selector name, used as the index of selectorNames in storageGroup . Therefore, the name of each selector must be unique.
storageType	String	Specifies the storage type. Currently, only evs (EVS volumes) and local (local volumes) are supported. The local storage does not support disk selection. All local disks will form a VG. Therefore, only one storageSelector of the local type is allowed.
matchLabels	matchLabels object	Matching field of an EVS volume. The size , volumeType , metadataEncrypted , metadataCmkid and count fields are supported.

Table 4-628 matchLabels

Parameter	Type	Description
size	String	Matched disk size. If this parameter is left unspecified, the disk size is not limited. Example: 100
volumeType	String	EVS disk type. SSD , GPSSD , SAS , ESSD , and SATA are supported. If this parameter is left blank, there is no restriction on the disk type.
metadataEncrypted	String	Disk encryption identifier. If this parameter is set to 0 , it specifies that the disk is not encrypted. If this parameter is set to 1 , it specifies that the disk is encrypted. If this parameter is left blank, there is no restriction on the disk encryption identifier.
metadataCmkid	String	CMK ID of an encrypted disk. The value is a string of 36 bytes. If this parameter is left blank, there is no restriction on the disk key ID.
count	String	Number of disks to be selected. If this parameter is left blank, all disks of this type are selected.

Table 4-629 StorageGroups

Parameter	Type	Description
name	String	Name of a virtual storage group, which must be unique. NOTE <ul style="list-style-type: none"> If cceManaged is set to true, the name must be set to vgpaas. If the data disk is used as a temporary storage volume, the name must be vg-everest-localvolume-ephemeral. If the data disk is used as a persistent storage volume, the name must be vg-everest-localvolume-persistent.
cceManaged	Boolean	Storage space for Kubernetes and runtime components. Only one group can be set to true . If this parameter is left blank, the default value false is used.
selectorNames	Array of strings	This parameter corresponds to name in storageSelectors . A group can match multiple selectors, but a selector can match only one group.

Parameter	Type	Description
virtualSpaces	Array of VirtualSpace objects	Detailed management of space configuration in a group.

Table 4-630 VirtualSpace

Parameter	Type	Description
name	String	Name of a virtualSpace. Options: share , kubernetes , runtime , and user <ul style="list-style-type: none"> • share: shared space configuration (with the runtime and Kubernetes partitions cancelled). lvmConfig needs to be configured. • Kubernetes: Kubernetes space configuration. lvmConfig needs to be configured. • runtime: runtime space configuration. runtimeConfig needs to be configured. • user: user space configuration. lvmConfig needs to be configured.
size	String	Size of a virtualSpace. The value must be an integer in percentage. Example: 90%. NOTE The sum of the percentages of all virtualSpaces in a group cannot exceed 100%.
lvmConfig	LVMConfig object	LVM configuration managements, applicable to the <i>share</i> *, kubernetes , and user spaces. Note that one virtual space supports only one config.
runtimeConfig	RuntimeConfig object	runtime configurations, applicable to the runtime space. Note that one virtual space supports only one config.

Table 4-631 LVMConfig

Parameter	Type	Description
lvType	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.

Parameter	Type	Description
path	String	Path to which the disk is attached. This parameter takes effect only in user configuration. The value is an absolute path. Digits, letters, periods (.), hyphens (-), and underscores (_) are allowed.

Table 4-632 RuntimeConfig

Parameter	Type	Description
lvType	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.

Table 4-633 NodePublicIP

Parameter	Type	Description
ids	Array of strings	IDs of existing EIPs. The quantity cannot be greater than the number of nodes to be created. NOTE If ids has been set, you do not need to set count and eip .
count	Integer	Number of EIPs to be dynamically created. NOTE count and eip must be set at the same time.
eip	NodeEIPSpec object	EIP configuration.

Table 4-634 NodeEIPSpec

Parameter	Type	Description
iptype	String	EIP type, specified in publicip.type in the API for assigning an EIP. For details, see Assigning an EIP .
bandwidth	NodeBandwidth object	Bandwidth parameters of the EIP

Table 4-635 NodeBandwidth

Parameter	Type	Description
chargemode	String	<p>Bandwidth billing mode.</p> <ul style="list-style-type: none"> • If this field is not specified, the billing is based on bandwidth. • If the field is null, the billing is based on bandwidth. • If the field value is traffic, the billing is based on traffic. • If the value is out of the preceding options, the cloud server will fail to be created. <p>NOTE</p> <ul style="list-style-type: none"> • Billed by bandwidth: The billing will be based on the data transmission rate (in Mbps) of public networks. This billing mode is recommended if your bandwidth usage is higher than 10%. • Billed by traffic: The billing is based on the total amount of data (in GB) transmitted over the public network. This mode is available only when you are creating a pay-per-use node. This billing mode is recommended if your bandwidth usage is lower than 10%.
size	Integer	<p>Bandwidth size, specified in bandwidth.size in the API for assigning an EIP.</p> <p>For details, see Assigning an EIP.</p>
sharetype	String	<p>Bandwidth sharing type. Value options: PER (exclusive bandwidth)</p>

Table 4-636 NodeNicSpec

Parameter	Type	Description
primaryNic	NicSpec object	Description of the primary NIC.
extNics	Array of NicSpec objects	<p>Extension NIC</p> <p>NOTE This parameter is not supported when you add a node to a node pool.</p>

Table 4-637 NicSpec

Parameter	Type	Description
subnetId	String	Network ID of the subnet to which a NIC belongs. If subnetId is not specified when a primary NIC is creating, the cluster subnet will be used. If subnetList is also configured for a node pool, the subnetList field is used for adding subnets to the node pool. When creating an extension NIC, you must specify subnetId .
fixedIps	Array of strings	The IP address of the primary ENI is specified using fixedIps . The number of IP addresses cannot be greater than the number of created nodes. Either fixedIps or ipBlock can be specified. fixedIps cannot be specified for extension ENIs.
ipBlock	String	CIDR format of the primary NIC IP range. The IP address of the created node falls in this range. fixedIps and ipBlock cannot be specified at the same time.

Table 4-638 Taint

Parameter	Type	Description
key	String	Key.
value	String	Value.
effect	String	Effect.

Table 4-639 UserTag

Parameter	Type	Description
key	String	Key of the cloud server label. The value cannot start with CCE- or __type_baremetal .
value	String	Value of the cloud server label.

Table 4-640 Runtime

Parameter	Type	Description
name	String	<p>Container runtime. Default value:</p> <ul style="list-style-type: none"> Clusters earlier than v1.25: The default value is docker. Clusters of v1.25 or later: The default container runtime varies depending on the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker; for nodes running other OSs, the default container runtime is containerd.

Table 4-641 NodeExtendParam

Parameter	Type	Description
ecs:performancetype	String	ECS flavor types. This field is returned in the response.
orderId	String	Order ID. This parameter is returned in the response when the node is billed on a yearly/monthly basis with auto payment enabled (only in creation scenarios).
productId	String	Product ID. This field is returned in the response when the node is yearly/monthly-billed with auto payment enabled.
maxPods	Integer	<p>Maximum number of pods that can be created on a node, including the default system pods. Value range: 16 to 256.</p> <p>This limit prevents the node from being overloaded of pods.</p> <p>The number of pods that can be created on a node is determined by multiple parameters. For details, see Maximum Number of Pods That Can Be Created on a Node.</p>
periodType	String	<ul style="list-style-type: none"> month: The unit is month. year: The unit is year. <p>NOTE</p> <p>Request parameter, which is valid and mandatory only when billingMode is set to 1 (yearly/monthly billing) or 2 (yearly/monthly billing with auto renewal enabled) (discarded).</p> <p>Response parameter, which is returned only when a yearly/monthly node is created.</p>

Parameter	Type	Description
periodNum	Integer	<p>Subscription duration. The value can be:</p> <ul style="list-style-type: none"> • If periodType is month, the value ranges from 1 to 9. • If periodType is year, the value is 1. <p>NOTE Request parameter, which is valid and mandatory when billingMode is set to 1 or 2 (discarded). Response parameter, which is returned only when a yearly/monthly node is created.</p>
isAutoRenew	String	<p>Whether auto renewal is enabled.</p> <ul style="list-style-type: none"> • true: Auto renewal is enabled. • false: Auto renewal is not enabled. <p>NOTE This field is valid when billingMode is set to 1 or 2 (discarded). If not specified, auto renewal is not enabled.</p>
isAutoPay	String	<p>Whether to deduct fees automatically.</p> <ul style="list-style-type: none"> • true: Enable automatic fee deduction. • false: Do not enable automatic fee deduction. <p>NOTE This field is valid when billingMode is set to 1 or 2 (discarded). If not specified when billingMode is set to 1, auto fee deduction is not enabled. (Discarded: If not specified when billingMode is set to 2, auto fee deduction is enabled.)</p>

Parameter	Type	Description
DockerLVMConfigOverride	String	<p>Docker data disk configuration item. (This parameter has been discarded. Use the storage field instead.) Example default configuration: "DockerLVMConfigOverride":"dockerThinpool=vgpaas/90%VG;kubernetesLV=vgpaas/10%VG;diskType=evs;lvType=linear"</p> <p>By default, if no VD disk is available, an error occurs because the data disk fails to be found. Set diskType based on the actual drive letter type.</p> <p>The following fields are included:</p> <ul style="list-style-type: none"> • userLV (optional): size of the user space, for example, vgpaas/20%VG. • userPath (optional): mount path of the user space, for example, /home/wqt-test. • diskType: disk type. Currently, only evs, hdd, and ssd are supported. • lvType: type of a logic volume. The value can be linear or striped. • dockerThinpool: Docker space size, for example, vgpaas/60%VG. • kubernetesLV: kubelet space size, for example, vgpaas/20%VG.
dockerBaseSize	Integer	<p>Available disk space of a single container on a node, in GB. This parameter has been discarded. Use containerBaseSize.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In Device Mapper mode, the default value is 10. In OverlayFS mode, the available space of a single container is not limited by default, and the dockerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters of the new versions.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that dockerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device.</p>

Parameter	Type	Description
containerBaseSize	Integer	<p>Available disk space of a single container on a node, in GB.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In OverlayFS mode, the available space of a single container is not limited by default. In Device Mapper mode, the default value is 10, and the containerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters v1.23.14-r0, v1.25.9-r0, v1.27.6-r0, v1.28.4-r0 or later.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that containerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device. In new versions, the Device Mapper mode is used only by BMSs in the same resource pool and is being discarded.</p>
publicKey	String	Public key of a node.
alpha.cce/preInstall	String	<p>Pre-installation script.</p> <p>NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)</p>
alpha.cce/postInstall	String	<p>Post-installation script.</p> <p>NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)</p>
alpha.cce/NodeImageID	String	This parameter is required when a custom image is used to create a BMS node.

Parameter	Type	Description
nicMultiqueue	String	<ul style="list-style-type: none"> Number of ENI queues. Example setting: <code>"[{"queue":4}]"</code> The following fields are included: <ul style="list-style-type: none"> queue: number of ENI queues. This field can be configured only for BMS nodes in a CCE Turbo cluster. Supported proportions are {"1":128, "2":92, "4":92, "8":32, "16":16, "28":9}. That is, if there is one queue, a maximum of 128 ENIs can be bound. If there are two queues, a maximum of 92 ENIs can be bound for two queues. A larger number of ENI queues indicates higher performance but fewer ENIs can be bound. The queue settings cannot be changed after the node pool is created.
nicThreshold	String	<ul style="list-style-type: none"> ENI pre-binding thresholds. Example setting: <code>"0.3:0.6"</code> <ul style="list-style-type: none"> Low threshold (L): determines the minimum number of pre-bound ENIs (Min). Formula: $Min = Total\ number\ of\ ENIs\ of\ the\ node \times L$ High threshold (H): determines the maximum number of pre-bound ENIs (Max). Formula: $Max = Total\ number\ of\ ENIs\ of\ the\ node \times H$ Number of ENIs bound to a BMS node (B) and number of ENIs being used by pods (U): $U + Min < B < U + Max$ If the number of pre-bound ENIs on a BMS node is smaller than the minimum allowed, the system will bind more ENIs to make the numbers equal. If the number of pre-bound ENIs on a BMS node is larger than the maximum allowed, the system periodically unbinds ENIs (about every 2 minutes) to make the numbers equal. Both the thresholds are one-decimal-place values ranging from 0.0 to 1.0. The low threshold must be smaller than or equal to the high one. This field can be configured only for BMS nodes in a CCE Turbo cluster. Pre-binding ENIs can speed up workload creation but occupies IP addresses.

Parameter	Type	Description
chargingMode	Integer	Billing mode of a node. This parameter has been deprecated. Use the billingMode parameter in NodeSpec .
agency_name	String	Name of an agency An agency is created by a tenant administrator on Identity and Access Management (IAM) to provide temporary credentials for CCE nodes to access cloud servers. This parameter is returned only when it is transferred during node creation.
kubeReservedMem	Integer	Reserved node memory, which is reserved for Kubernetes components.
systemReservedMem	Integer	Reserved node memory, which is reserved for system components.
init-node-password	String	Node password. If this parameter is used as a response parameter, asterisks (*) will be displayed.

Table 4-642 HostnameConfig

Parameter	Type	Description
type	String	Configuration type of the Kubernetes node name. The default value is privatelp . <ul style="list-style-type: none"> • privatelp: The Kubernetes node is named after its IP address. • cceNodeName: The Kubernetes node is named after the CCE node. NOTE <ul style="list-style-type: none"> • For a node which is configured using cceNodeName, the name is the same as the Kubernetes node name and the ECS name. The node name cannot be changed. If the ECS name is changed on the ECS console, the node name will retain unchanged after ECS synchronization. • For a node which is configured using cceNodeName, to avoid a conflict between Kubernetes nodes, the system automatically adds a suffix to each node name. The suffix is in the format of A hyphen (-) Five random characters. The value of the random characters is a lowercase letter or a digit ranging from 0 to 9.

Table 4-643 NodePoolNodeAutoscaling

Parameter	Type	Description
enable	Boolean	Whether to enable auto scaling.
minNodeCount	Integer	Minimum number of nodes allowed if auto scaling is enabled. The value cannot be greater than the maximum number of nodes allowed by the cluster specifications.
maxNodeCount	Integer	Maximum number of nodes allowed if auto scaling is enabled. This value must be greater than or equal to the value of minNodeCount and cannot exceed the maximum number of nodes in the cluster specifications.
scaleDownCooldownTime	Integer	Interval between two scaling operations, in minutes. During this period, nodes added after a scale-up will not be deleted.
priority	Integer	Weight of a node pool. A node pool with a higher weight has a higher priority during scaling.

Table 4-644 NodeManagement

Parameter	Type	Description
serverGroupReference	String	Cloud server group ID. If this field is specified, all nodes in the node pool will be created in this group. The group ID can be specified only when you create the node pool and cannot be modified. When you specify a cloud server group, the number of nodes in the node pool cannot exceed the group quota.

Table 4-645 SecurityID

Parameter	Type	Description
id	String	Security group ID

Table 4-646 UpdateNodePoolStatus

Parameter	Type	Description
currentNode	Integer	Total number of nodes in the current node pool (excluding the nodes that are being deleted).

Parameter	Type	Description
creatingNode	Integer	Number of nodes that are being created in the node pool.
deletingNode	Integer	Number of nodes that are being deleted in the current node pool.
phase	String	<p>Node pool status. Options:</p> <ul style="list-style-type: none"> • Null: The node pool is available. (The number of nodes in the current node pool has reached the expected value, and no node scaling is being performed.) • Synchronizing: The node pool is being scaled. (The number of nodes in the current node pool has not reached the expected value, and no node scaling is being performed.) • Synchronized: The node pool scaling is pending. (The number of nodes in the current node pool has not reached the expected value, or node scaling is being performed.) • SoldOut: The node pool cannot be scaled out. (This field is used in multiple scenarios, for example, resources in the node pool have been sold out or the resource quota is insufficient.) <p>NOTE This node pool status has been deprecated. It is reserved only for compatibility. Do not use it anymore. Use the following instead:</p> <ul style="list-style-type: none"> • Node pool scaling status: You can obtain the status of the current node pool using parameters such as currentNode, creatingNode, or deletingNode. • Node pool scale-out: You can use conditions to obtain the detailed status of a node pool. Scalable can replace SoldOut. • Deleting: The node pool is being deleted. • Error: An error occurred in the node pool.
conditions	Array of NodePoolCondition objects	Node pool status details. For details, see the definition of Condition .

Table 4-647 NodePoolCondition

Parameter	Type	Description
type	String	Condition type. The options are as follows: <ul style="list-style-type: none"> • Scalable: whether a node pool can be scaled. If the status is False, node pool scaling will not be triggered again. • QuotaInsufficient: Quotas on which node pool scaling depends are insufficient, affecting the node pool scaling status. • ResourceInsufficient: Resources on which node pool scaling depends are insufficient, affecting the node pool scaling status. • UnexpectedError: The node pool fails to be scaled out due to unexpected reasons, affecting the node pool scaling status. • LockedByOrder: The yearly/monthly-billed node pool is locked due to an order. The value of Reason is the ID of the pending order. • Error: A node pool error occurs. A common trigger is deletion failure.
status	String	Current status of Condition . The options are as follows: <ul style="list-style-type: none"> • "True" • "False"
lastProbeTime	String	Time when the status was last checked
lastTransitTime	String	Time when the status was last changed
reason	String	Reason why the status was last changed
message	String	Detailed condition description

Example Requests

- Change the number of nodes in the node pool to 1.
PUT /api/v3/projects/{project_id}/clusters/{cluster_id}/nodepools/{nodepool_id}

```
{
  "metadata" : {
    "name" : "lc-it-nodepool-3"
  },
  "spec" : {
    "nodeTemplate" : {
      "k8sTags" : { },
      "taints" : [ {
        "key" : "status",
        "value" : "unavailable",
        "effect" : "NoSchedule"
      } ],
    }
  }
}
```

```

    "userTags" : [ ]
  },
  "autoscaling" : {
    "enable" : false,
    "minNodeCount" : 0,
    "maxNodeCount" : 0,
    "scaleDownCooldownTime" : 0,
    "priority" : 0
  },
  "initialNodeCount" : 1
}
}

```

- **Modify the key pair.**

PUT /api/v3/projects/{project_id}/clusters/{cluster_id}/nodepools/{nodepool_id}

```

{
  "metadata" : {
    "name" : "lc-it-nodepool-2"
  },
  "spec" : {
    "nodeTemplate" : {
      "k8sTags" : { },
      "taints" : [ {
        "key" : "status",
        "value" : "unavailable",
        "effect" : "NoSchedule"
      } ],
      "userTags" : [ ],
      "login" : {
        "sshKey" : "KeyPair-IES"
      }
    },
    "autoscaling" : {
      "enable" : false,
      "minNodeCount" : 0,
      "maxNodeCount" : 0,
      "scaleDownCooldownTime" : 0,
      "priority" : 0
    },
    "initialNodeCount" : 1
  }
}

```

Example Responses

Status code: 200

Information about the specified node pool is successfully updated.

```

{
  "kind" : "NodePool",
  "apiVersion" : "v3",
  "metadata" : {
    "name" : "lc-it-nodepool-3",
    "uid" : "1deef848-690d-11ea-a11b-0255ac1001b7"
  },
  "spec" : {
    "initialNodeCount" : 1,
    "type" : "vm",
    "nodeTemplate" : {
      "flavor" : "Sit3.xlarge.2",
      "az" : "*****",
      "os" : "EulerOS 2.5",
      "login" : {
        "sshKey" : "KeyPair-001"
      }
    },
    "rootVolume" : {
      "volumetype" : "SAS",

```



```
"size" : 40
},
"dataVolumes" : [ {
  "volumetype" : "SAS",
  "size" : 100,
  "extendParam" : {
    "useType" : "docker"
  }
}],
"publicIP" : {
  "eip" : {
    "bandwidth" : { }
  }
},
"nodeNicSpec" : {
  "primaryNic" : {
    "subnetId" : "7e767d10-7548-4df5-ad72-aeac1d08bd8a"
  }
},
"billingMode" : 0,
"extendParam" : {
  "maxPods" : 110
},
"k8sTags" : {
  "cce.cloud.com/cce-nodepool" : "lc-it-nodepool-3"
}
},
"autoscaling" : { },
"nodeManagement" : { }
},
"status" : {
  "phase" : ""
}
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

- Change the number of nodes in the node pool to 1.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

import java.util.List;
import java.util.ArrayList;
import java.util.Map;
import java.util.HashMap;

public class UpdateNodePoolSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
```

```
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

CceClient client = CceClient.newBuilder()
    .withCredential(auth)
    .withRegion(CceRegion.valueOf("<YOUR REGION>"))
    .build();
UpdateNodePoolRequest request = new UpdateNodePoolRequest();
request.withClusterId("{cluster_id}");
request.withNodepoolId("{nodepool_id}");
NodePoolUpdate body = new NodePoolUpdate();
NodePoolNodeAutoscaling autoscalingSpec = new NodePoolNodeAutoscaling();
autoscalingSpec.withEnable(false)
    .withMinNodeCount(0)
    .withMaxNodeCount(0)
    .withScaleDownCooldownTime(0)
    .withPriority(0);
List<Taint> listNodeTemplateTaints = new ArrayList<>();
listNodeTemplateTaints.add(
    new Taint()
        .withKey("status")
        .withValue("unavailable")
        .withEffect(Taint.EffectEnum.fromValue("NoSchedule"))
);
NodeSpecUpdate nodeTemplateSpec = new NodeSpecUpdate();
nodeTemplateSpec.withTaints(listNodeTemplateTaints);
NodePoolSpecUpdate specbody = new NodePoolSpecUpdate();
specbody.withNodeTemplate(nodeTemplateSpec)
    .withInitialNodeCount(1)
    .withAutoscaling(autoscalingSpec);
NodePoolMetadataUpdate metadatabody = new NodePoolMetadataUpdate();
metadatabody.withName("lc-it-nodepool-3");
body.withSpec(specbody);
body.withMetadata(metadatabody);
request.withBody(body);
try {
    UpdateNodePoolResponse response = client.updateNodePool(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

- **Modify the key pair.**

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

import java.util.List;
```

```
import java.util.ArrayList;
import java.util.Map;
import java.util.HashMap;

public class UpdateNodePoolSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();

        UpdateNodePoolRequest request = new UpdateNodePoolRequest();
        request.withClusterId("{cluster_id}");
        request.withNodepoolId("{nodepool_id}");
        NodePoolUpdate body = new NodePoolUpdate();
        NodePoolNodeAutoscaling autoscalingSpec = new NodePoolNodeAutoscaling();
        autoscalingSpec.withEnable(false)
            .withMinNodeCount(0)
            .withMaxNodeCount(0)
            .withScaleDownCooldownTime(0)
            .withPriority(0);
        Login loginNodeTemplate = new Login();
        loginNodeTemplate.withSshKey("KeyPair-IES");
        List<Taint> listNodeTemplateTaints = new ArrayList<>();
        listNodeTemplateTaints.add(
            new Taint()
                .withKey("status")
                .withValue("unavailable")
                .withEffect(Taint.EffectEnum.fromValue("NoSchedule"))
        );
        NodeSpecUpdate nodeTemplateSpec = new NodeSpecUpdate();
        nodeTemplateSpec.withTaints(listNodeTemplateTaints)
            .withLogin(loginNodeTemplate);
        NodePoolSpecUpdate specbody = new NodePoolSpecUpdate();
        specbody.withNodeTemplate(nodeTemplateSpec)
            .withInitialNodeCount(1)
            .withAutoscaling(autoscalingSpec);
        NodePoolMetadataUpdate metadatabody = new NodePoolMetadataUpdate();
        metadatabody.withName("lc-it-nodepool-2");
        body.withSpec(specbody);
        body.withMetadata(metadatabody);
        request.withBody(body);
        try {
            UpdateNodePoolResponse response = client.updateNodePool(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
        }
    }
}
```

```
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

- Change the number of nodes in the node pool to 1.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateNodePoolRequest()
        request.cluster_id = "{cluster_id}"
        request.nodepool_id = "{nodepool_id}"
        autoscalingSpec = NodePoolNodeAutoscaling(
            enable=False,
            min_node_count=0,
            max_node_count=0,
            scale_down_cooldown_time=0,
            priority=0
        )
        listTaintsNodeTemplate = [
            Taint(
                key="status",
                value="unavailable",
                effect="NoSchedule"
            )
        ]
        nodeTemplateSpec = NodeSpecUpdate(
            taints=listTaintsNodeTemplate
        )
        specbody = NodePoolSpecUpdate(
            node_template=nodeTemplateSpec,
            initial_node_count=1,
            autoscaling=autoscalingSpec
        )
        metadatabody = NodePoolMetadataUpdate(
            name="lc-it-nodepool-3"
        )
        request.body = NodePoolUpdate(
            spec=specbody,
            metadata=metadatabody
        )
        response = client.update_node_pool(request)
        print(response)
    except exceptions.ClientRequestException as e:
```

```
print(e.status_code)
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

- **Modify the key pair.**

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateNodePoolRequest()
        request.cluster_id = "{cluster_id}"
        request.nodepool_id = "{nodepool_id}"
        autoscalingSpec = NodePoolNodeAutoscaling(
            enable=False,
            min_node_count=0,
            max_node_count=0,
            scale_down_cooldown_time=0,
            priority=0
        )
        loginNodeTemplate = Login(
            ssh_key="KeyPair-IES"
        )
        listTaintsNodeTemplate = [
            Taint(
                key="status",
                value="unavailable",
                effect="NoSchedule"
            )
        ]
        nodeTemplateSpec = NodeSpecUpdate(
            taints=listTaintsNodeTemplate,
            login=loginNodeTemplate
        )
        specbody = NodePoolSpecUpdate(
            node_template=nodeTemplateSpec,
            initial_node_count=1,
            autoscaling=autoscalingSpec
        )
        metadatabody = NodePoolMetadataUpdate(
            name="lc-it-nodepool-2"
        )
        request.body = NodePoolUpdate(
            spec=specbody,
            metadata=metadatabody
        )
        response = client.update_node_pool(request)
        print(response)
```

```
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

- Change the number of nodes in the node pool to 1.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateNodePoolRequest{}
    request.ClusterId = "{cluster_id}"
    request.NodepoolId = "{nodepool_id}"
    enableAutoscaling := false
    minNodeCountAutoscaling := int32(0)
    maxNodeCountAutoscaling := int32(0)
    scaleDownCooldownTimeAutoscaling := int32(0)
    priorityAutoscaling := int32(0)
    autoscalingSpec := &model.NodePoolNodeAutoscaling{
        Enable: &enableAutoscaling,
        MinNodeCount: &minNodeCountAutoscaling,
        MaxNodeCount: &maxNodeCountAutoscaling,
        ScaleDownCooldownTime: &scaleDownCooldownTimeAutoscaling,
        Priority: &priorityAutoscaling,
    }
    valueTaints := "unavailable"
    var listTaintsNodeTemplate = []model.Taint{
        {
            Key: "status",
            Value: &valueTaints,
            Effect: model.GetTaintEffectEnum().NO_SCHEDULE,
        },
    }
    nodeTemplateSpec := &model.NodeSpecUpdate{
        Taints: listTaintsNodeTemplate,
    }
    specbody := &model.NodePoolSpecUpdate{
```

```

    NodeTemplate: nodeTemplateSpec,
    InitialNodeCount: int32(1),
    Autoscaling: autoscalingSpec,
  }
  metadatabody := &model.NodePoolMetadataUpdate{
    Name: "lc-it-nodepool-3",
  }
  request.Body = &model.NodePoolUpdate{
    Spec: specbody,
    Metadata: metadatabody,
  }
  response, err := client.UpdateNodePool(request)
  if err == nil {
    fmt.Printf("%+v\n", response)
  } else {
    fmt.Println(err)
  }
}

```

- **Modify the key pair.**

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateNodePoolRequest{}
    request.ClusterId = "{cluster_id}"
    request.NodepoolId = "{nodepool_id}"
    enableAutoscaling := false
    minNodeCountAutoscaling := int32(0)
    maxNodeCountAutoscaling := int32(0)
    scaleDownCooldownTimeAutoscaling := int32(0)
    priorityAutoscaling := int32(0)
    autoscalingSpec := &model.NodePoolNodeAutoscaling{
        Enable: &enableAutoscaling,
        MinNodeCount: &minNodeCountAutoscaling,
        MaxNodeCount: &maxNodeCountAutoscaling,
        ScaleDownCooldownTime: &scaleDownCooldownTimeAutoscaling,
        Priority: &priorityAutoscaling,
    }
    sshKeyLogin := "KeyPair-IES"
    loginNodeTemplate := &model.Login{

```

```

    SshKey: &sshKeyLogin,
  }
  valueTaints:= "unavailable"
  var listTaintsNodeTemplate = []model.Taint{
    {
      Key: "status",
      Value: &valueTaints,
      Effect: model.GetTaintEffectEnum().NO_SCHEDULE,
    },
  }
  nodeTemplateSpec := &model.NodeSpecUpdate{
    Taints: listTaintsNodeTemplate,
    Login: loginNodeTemplate,
  }
  specbody := &model.NodePoolSpecUpdate{
    NodeTemplate: nodeTemplateSpec,
    InitialNodeCount: int32(1),
    Autoscaling: autoscalingSpec,
  }
  metadatabody := &model.NodePoolMetadataUpdate{
    Name: "lc-it-nodepool-2",
  }
  request.Body = &model.NodePoolUpdate{
    Spec: specbody,
    Metadata: metadatabody,
  }
  response, err := client.UpdateNodePool(request)
  if err == nil {
    fmt.Printf("%+v\n", response)
  } else {
    fmt.Println(err)
  }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Information about the specified node pool is successfully updated.

Error Codes

See [Error Codes](#).

4.4.5 Deleting a Node Pool

Function

This API is used to delete a specified node pool.

NOTE

The URL for cluster management is in the format of https://Endpoint/uri. In the URL, **uri** indicates the resource path, that is, the path for API access.

Calling Method

For details, see [Calling APIs](#).

URI

DELETE /api/v3/projects/{project_id}/clusters/{cluster_id}/nodepools/{nodepool_id}

Table 4-648 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
nodepool_id	Yes	String	Node pool ID.

Request Parameters

Table 4-649 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-650 Response body parameters

Parameter	Type	Description
kind	String	API type. The value is fixed at NodePool .

Parameter	Type	Description
apiVersion	String	API version. The value is fixed at v3 .
metadata	NodePoolMetadata object	Metadata of the node pool.
spec	NodePoolSpec object	Node pool specifications.
status	DeleteNodePoolStatus object	Node pool status.

Table 4-651 NodePoolMetadata

Parameter	Type	Description
name	String	Node pool name. NOTE Naming rules: Enter 1 to 50 characters, starting with a lowercase letter and not ending with a hyphen (-). Only lowercase letters, digits, and hyphens (-) are allowed. <ul style="list-style-type: none"> You cannot create node pools named DefaultPool.
uid	String	UID of the node pool. The value is automatically generated after the object is updated. A user-defined value will not take effect.
annotations	Map<String,String>	Annotations of a node pool in key-value pairs. This parameter is used only for query and cannot be input through a request. Entered data of this parameter is invalid.
updateTimestamp	String	Update time.
creationTimestamp	String	Creation time.

Table 4-652 NodePoolSpec

Parameter	Type	Description
type	String	Node pool type. If this parameter is left blank, the value vm is used by default. <ul style="list-style-type: none"> • vm: ECS • ElasticBMS: C6 general computing-plus BMS. An example flavor is c6.22xlarge.2.physical. • bms: BMS
nodeTemplate	NodeSpec object	Detailed parameters of the node pool template.
initialNodeCount	Integer	Initial number of nodes for the node pool. When queried, the value is the number of target nodes in the node pool.
autoscaling	NodePoolNodeAutoscaling object	Auto scaling parameters
nodeManagement	NodeManagement object	Node management configuration
podSecurityGroups	Array of SecurityID objects	Security group configuration. This parameter is available only for CCE Turbo clusters.
customSecurityGroups	Array of strings	Custom security group settings for a node pool. New nodes scaled out in a node pool can be bound to a specified security group. <ul style="list-style-type: none"> • Specifying no security group ID will add the new nodes to the default security group of the worker nodes. • Specifying a valid security group ID will put new nodes in that security group. • When specifying a security group, do not modify the rules of the port on which CCE running depends. For details, see How Do I Harden the Automatically Created Security Group Rules for CCE Cluster Nodes.

Table 4-653 NodeSpec

Parameter	Type	Description
flavor	String	Node specifications. For details about the node specifications supported by CCE, see Node Flavor Description .
az	String	AZ where the node to be created is located. You need to specify the AZ name. If it is set to random , the node will be created in a random AZ. For details about AZs supported by CCE, see Regions and Endpoints .
os	String	Node OS. For details about the supported OSs, see Node OS . NOTE <ul style="list-style-type: none"> The system automatically selects the supported OS based on the cluster version. If the current cluster version does not support the OS, an error will be reported. If alpha.cce/NodeImageID in extendParam is specified during node creation, you do not need to configure this parameter. This parameter is mandatory when creating a node pool. If you are using shared disk space when creating a node, this parameter is mandatory. Specifically, the disk initialization configuration management parameter storage is used and the name field of virtualSpaces in StorageGroups is set to share.
login	Login object	Node login mode. Either the key pair or password must be used for login.
rootVolume	Volume object	Information about disks on the node
dataVolumes	Array of Volume objects	Data disk parameters of the node. Currently, you can add the second data disk for your node on the CCE console. This data disk is used by the container runtime and kubelet. Do not uninstall this disk. Otherwise, the node will become unavailable. For DeC nodes, the parameter description is the same as that for rootVolume .

Parameter	Type	Description
storage	Storage object	<p>Disk initialization management parameter. This parameter is complex to configure. For details, see Attaching Disks to a Node.</p> <p>If this parameter retains its default, disks are managed based on the DockerLVMConfigOverride (discarded) parameter in extendParam. This parameter is supported by clusters of version 1.15.11 and later.</p> <p>NOTE</p> <ul style="list-style-type: none"> • If a node specification involves both local disks and EVS disks, do not retain the default value of this parameter to prevent unexpected disk partitions. • If you want to change the value range of a data disk to 20 to 32768, do not use the default value. • If you want to use the shared disk space (with the runtime and Kubernetes partitions cancelled), do not retain the default value of this parameter. For details about the shared disk space, see Data Disk Space Allocation.
publicIP	NodePublicIP object	<p>EIP of a node.</p> <p>NOTE This parameter is not supported when you add a node to a node pool.</p>
nodeNicSpec	NodeNicSpec object	NIC of the node
count	Integer	Number of nodes to be created in a batch. The value must be a positive integer greater than or equal to 1 and less than or equal to the defined limit. This parameter can be left blank when it is used for a node pool.
billingMode	Integer	<p>Node billing mode.</p> <ul style="list-style-type: none"> • 0: pay-per-use • 1: yearly/monthly • 2: (discarded) yearly/monthly billing with auto payment enabled

Parameter	Type	Description
taints	Array of Taint objects	<p>You can add taints to created nodes to configure anti-affinity. A maximum of 20 taints can be added. Each taint contains the following parameters:</p> <ul style="list-style-type: none"> • Key: A key must contain 1 to 63 characters starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain name can be used as the prefix of a key. • Value: A value must start with a letter or digit and can contain a maximum of 63 characters, including letters, digits, hyphens (-), underscores (_), and periods (.). • Effect: Available options are NoSchedule, PreferNoSchedule, and NoExecute. When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query. <p>Example:</p> <pre> taints: [{ "key": "status", "value": "unavailable", "effect": "NoSchedule" }, { "key": "looks", "value": "bad", "effect": "NoSchedule" }] </pre>

Parameter	Type	Description
k8sTags	Map<String,String>	<p>Defined in key-value pairs. A maximum of 20 key-value pairs are allowed.</p> <ul style="list-style-type: none"> Key: Enter 1 to 63 characters, starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain can be prefixed to a key and contain a maximum of 253 characters. Example DNS subdomain: example.com/my-key Value: The value can be left blank or contain 1 to 63 characters that start with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed in the character string. When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query. <p>Example:</p> <pre>"k8sTags": { "key": "value" }</pre>
ecsGroupId	String	<p>ECS group ID. If this parameter is configured, nodes will be created in the specified ECS group.</p> <p>NOTE This configuration does not take effect when a node pool is created. To ensure that all nodes in a node pool are in the same ECS group, configure nodeManagement in the node pool.</p>
dedicatedHostId	String	<p>ID of the DeH to which the node is scheduled.</p> <p>NOTE This parameter is not supported when you add a node during node pool creation.</p>

Parameter	Type	Description
userTags	Array of UserTag objects	<p>Cloud server tag. The key of a tag must be unique. The maximum number of custom tags supported by CCE depends on the region and cannot exceed 8.</p> <p>When creating a node, you can specify the initial value for this parameter and this field is not returned during query. In using node pools, the initial value can be specified in the node template and this field is returned during query. In other scenarios, this field is not returned during query.</p> <p>NOTE Use only letters, Unicode characters, digits, special characters(-, _). Max characters: 36</p>
runtime	Runtime object	<p>Container runtime:</p> <ul style="list-style-type: none"> • Clusters of v1.25 or earlier: docker. • Clusters of v1.25 or later: Container runtime varies with the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker. For nodes running other OSs, the default container runtime is containerd.

Parameter	Type	Description
initializedConditions	Array of strings	<p>Custom initialization flag, which is left blank by default.</p> <p>Before CCE nodes are initialized, node.cloudprovider.kubernetes.io/uninitialized is added to the nodes to prevent pods from being scheduled to these nodes. When creating a node, you can configure the initializedConditions parameter to control the taint removal time. By default, the timeout period is not configured.</p> <p>Example:</p> <ol style="list-style-type: none"> 1. Create a node and add parameter "initializedConditions": ["CCEInitial", "CustomedInitial"]. 2. After custom initialization is complete, CCE will call a Kubernetes API (for example, PATCH /v1/nodes/{node_ip}/status) to update the node conditions by adding two labels of types CCEInitial and CustomedInitial and setting the status to True, as shown in the following: <pre data-bbox="817 1055 1430 1205"> status: conditions: - type: CCEInitial status: 'True' - type: CustomedInitial status: 'True' </pre> <ol style="list-style-type: none"> 1. CCE polls status.Conditions of nodes to check whether there are conditions of types CCEInitial and CustomedInitial. If such conditions exist and the status is True, the node initialization is complete and the initialization taint is removed. 2. initializedConditions allows you to configure a timeout period during node creation. For example, "initializedConditions": ["CCEInitial:15m", "CustomedInitial:15m"] indicates that the timeout period is 15 minutes. After the timeout period is reached, the initialization conditions will be automatically ignored and the initialization taint will be removed when CCE polls the node. <ul style="list-style-type: none"> ● Use only letters and digits. Max. characters: 20. ● Max. flags: 2. ● The unit of the timeout period is minute (m).

Parameter	Type	Description
extendParam	NodeExtendParam object	Extended parameters for creating a node.
hostnameConfig	HostnameConfig object	Kubernetes node name configuration parameter, which is supported by clusters of v1.23.6-r0 to v1.25 or clusters of v1.25.2-r0 or later versions.

Table 4-654 Login

Parameter	Type	Description
sshKey	String	Name of the key pair used for login.
userPassword	UserPassword object	Password used for node login.

Table 4-655 UserPassword

Parameter	Type	Description
username	String	Login account. The default value is root .
password	String	<p>If a username and a password are used to create a node, this field is shielded in the response body.</p> <p>A password must meet the following complexity requirements:</p> <ul style="list-style-type: none"> • Contains 8 to 26 characters. • Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (!@\$%^-_=+[]{};,:./?~#*) • Cannot contain the username or the username spelled backwards. <p>The password field must be salted during node creation. For details, see Adding a Salt in the password Field When Creating a Node.</p>

Table 4-656 Volume

Parameter	Type	Description
size	Integer	Disk size, in GB. <ul style="list-style-type: none"> System disk: 40 to 1024 Value range for the first data disk: 20 to 32768 (When storage is not specified, this value range becomes 100 to 32768.) Value range for other data disks: 10 to 32768. (When storage is not specified, this value range becomes 100 to 32768.)
volumetype	String	Disk type. For details about possible values, see the description of the root_volume parameter in the API used to create an ECS. <ul style="list-style-type: none"> SAS: high I/O SAS disk SSD: ultra-high I/O SSD disk SATA: common I/O SATA disk SATA disks have been removed from EVS. You can find them attached only on existing nodes.
extendParam	Map<String,Object>	Extended disk parameters, defined in extendparam in the API used to create an ECS. For details, see Creating an ECS .
cluster_id	String	ID of the storage pool used by the ECS system disk. This field is used only for DeC clusters, which functions as dssPoolID , that is, the ID of the DSS storage pool. To obtain the value, see the ID field in the API used to obtain details of a DSS storage pool .
cluster_type	String	Storage class of the cloud server system disk. The value is always dss . This field is used only for DeC clusters.
hw:passthrough	Boolean	<ul style="list-style-type: none"> Pay attention to this field if your ECS is SDI-compliant. If the value of this field is true, an SCSI disk will be created. If the node pool type is ElasticBMS, this field must be set to true. If a node specification involves local disks and EVS disks at the same time, set the disk initialization parameters. For details, see Attaching Disks to a Node.
metadata	VolumeMetadata object	EVS disk encryption information. This field is mandatory only when you need to encrypt the system disk or data disks of the node to be created.

Table 4-657 VolumeMetadata

Parameter	Type	Description
<code>__system__encrypted</code>	String	Whether the EVS disk is encrypted. The value 0 indicates that the EVS disk is not encrypted, and the value 1 indicates that the EVS disk is encrypted. If this parameter is not specified, EVS disks will not be encrypted by default.
<code>__system__cmkid</code>	String	CMK ID, which indicates encryption in metadata . This field is used with __system__encrypted .

Table 4-658 Storage

Parameter	Type	Description
<code>storageSelectors</code>	Array of StorageSelectors objects	Disk selection. Matched disks are managed according to matchLabels and storageType .
<code>storageGroups</code>	Array of StorageGroups objects	A storage group consists of multiple storage devices. It is used to divide storage space.

Table 4-659 StorageSelectors

Parameter	Type	Description
<code>name</code>	String	Selector name, used as the index of selectorNames in storageGroup . Therefore, the name of each selector must be unique.
<code>storageType</code>	String	Specifies the storage type. Currently, only evs (EVS volumes) and local (local volumes) are supported. The local storage does not support disk selection. All local disks will form a VG. Therefore, only one storageSelector of the local type is allowed.
<code>matchLabels</code>	matchLabels object	Matching field of an EVS volume. The size , volumeType , metadataEncrypted , metadataCmkid and count fields are supported.

Table 4-660 matchLabels

Parameter	Type	Description
size	String	Matched disk size. If this parameter is left unspecified, the disk size is not limited. Example: 100
volumeType	String	EVS disk type. SSD, GPSSD, SAS, ESSD, and SATA are supported. If this parameter is left blank, there is no restriction on the disk type.
metadataEncrypted	String	Disk encryption identifier. If this parameter is set to 0 , it specifies that the disk is not encrypted. If this parameter is set to 1 , it specifies that the disk is encrypted. If this parameter is left blank, there is no restriction on the disk encryption identifier.
metadataCmkid	String	CMK ID of an encrypted disk. The value is a string of 36 bytes. If this parameter is left blank, there is no restriction on the disk key ID.
count	String	Number of disks to be selected. If this parameter is left blank, all disks of this type are selected.

Table 4-661 StorageGroups

Parameter	Type	Description
name	String	Name of a virtual storage group, which must be unique. NOTE <ul style="list-style-type: none"> If cceManaged is set to true, the name must be set to vgpaas. If the data disk is used as a temporary storage volume, the name must be vg-everest-localvolume-ephemeral. If the data disk is used as a persistent storage volume, the name must be vg-everest-localvolume-persistent.
cceManaged	Boolean	Storage space for Kubernetes and runtime components. Only one group can be set to true . If this parameter is left blank, the default value false is used.
selectorNames	Array of strings	This parameter corresponds to name in storageSelectors . A group can match multiple selectors, but a selector can match only one group.

Parameter	Type	Description
virtualSpaces	Array of VirtualSpace objects	Detailed management of space configuration in a group.

Table 4-662 VirtualSpace

Parameter	Type	Description
name	String	Name of a virtualSpace. Options: share , kubernetes , runtime , and user <ul style="list-style-type: none"> • share: shared space configuration (with the runtime and Kubernetes partitions cancelled). lvmConfig needs to be configured. • Kubernetes: Kubernetes space configuration. lvmConfig needs to be configured. • runtime: runtime space configuration. runtimeConfig needs to be configured. • user: user space configuration. lvmConfig needs to be configured.
size	String	Size of a virtualSpace. The value must be an integer in percentage. Example: 90%. NOTE The sum of the percentages of all virtualSpaces in a group cannot exceed 100%.
lvmConfig	LVMConfig object	LVM configuration managements, applicable to the <i>share</i> *, kubernetes , and user spaces. Note that one virtual space supports only one config.
runtimeConfig	RuntimeConfig object	runtime configurations, applicable to the runtime space. Note that one virtual space supports only one config.

Table 4-663 LVMConfig

Parameter	Type	Description
lvType	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.

Parameter	Type	Description
path	String	Path to which the disk is attached. This parameter takes effect only in user configuration. The value is an absolute path. Digits, letters, periods (.), hyphens (-), and underscores (_) are allowed.

Table 4-664 RuntimeConfig

Parameter	Type	Description
lvType	String	LVM write mode. linear indicates the linear mode. striped indicates the striped mode, in which multiple disks are used to form a strip to improve disk performance.

Table 4-665 NodePublicIP

Parameter	Type	Description
ids	Array of strings	IDs of existing EIPs. The quantity cannot be greater than the number of nodes to be created. NOTE If ids has been set, you do not need to set count and eip .
count	Integer	Number of EIPs to be dynamically created. NOTE count and eip must be set at the same time.
eip	NodeEIPSpec object	EIP configuration.

Table 4-666 NodeEIPSpec

Parameter	Type	Description
iptype	String	EIP type, specified in publicip.type in the API for assigning an EIP. For details, see Assigning an EIP .
bandwidth	NodeBandwidth object	Bandwidth parameters of the EIP

Table 4-667 NodeBandwidth

Parameter	Type	Description
chargemode	String	<p>Bandwidth billing mode.</p> <ul style="list-style-type: none"> • If this field is not specified, the billing is based on bandwidth. • If the field is null, the billing is based on bandwidth. • If the field value is traffic, the billing is based on traffic. • If the value is out of the preceding options, the cloud server will fail to be created. <p>NOTE</p> <ul style="list-style-type: none"> • Billed by bandwidth: The billing will be based on the data transmission rate (in Mbps) of public networks. This billing mode is recommended if your bandwidth usage is higher than 10%. • Billed by traffic: The billing is based on the total amount of data (in GB) transmitted over the public network. This mode is available only when you are creating a pay-per-use node. This billing mode is recommended if your bandwidth usage is lower than 10%.
size	Integer	<p>Bandwidth size, specified in bandwidth.size in the API for assigning an EIP.</p> <p>For details, see Assigning an EIP.</p>
sharetype	String	<p>Bandwidth sharing type. Value options: PER (exclusive bandwidth)</p>

Table 4-668 NodeNicSpec

Parameter	Type	Description
primaryNic	NicSpec object	Description of the primary NIC.
extNics	Array of NicSpec objects	<p>Extension NIC</p> <p>NOTE This parameter is not supported when you add a node to a node pool.</p>

Table 4-669 NicSpec

Parameter	Type	Description
subnetId	String	Network ID of the subnet to which a NIC belongs. If subnetId is not specified when a primary NIC is creating, the cluster subnet will be used. If subnetList is also configured for a node pool, the subnetList field is used for adding subnets to the node pool. When creating an extension NIC, you must specify subnetId .
fixedIps	Array of strings	The IP address of the primary ENI is specified using fixedIps . The number of IP addresses cannot be greater than the number of created nodes. Either fixedIps or ipBlock can be specified. fixedIps cannot be specified for extension ENIs.
ipBlock	String	CIDR format of the primary NIC IP range. The IP address of the created node falls in this range. fixedIps and ipBlock cannot be specified at the same time.

Table 4-670 Taint

Parameter	Type	Description
key	String	Key.
value	String	Value.
effect	String	Effect.

Table 4-671 UserTag

Parameter	Type	Description
key	String	Key of the cloud server label. The value cannot start with CCE- or __type_baremetal .
value	String	Value of the cloud server label.

Table 4-672 Runtime

Parameter	Type	Description
name	String	<p>Container runtime. Default value:</p> <ul style="list-style-type: none"> • Clusters earlier than v1.25: The default value is docker. • Clusters of v1.25 or later: The default container runtime varies depending on the OS. For nodes running EulerOS 2.5 or EulerOS 2.8, the default container runtime is docker; for nodes running other OSs, the default container runtime is containerd.

Table 4-673 NodeExtendParam

Parameter	Type	Description
ecs:performancetype	String	ECS flavor types. This field is returned in the response.
orderId	String	Order ID. This parameter is returned in the response when the node is billed on a yearly/monthly basis with auto payment enabled (only in creation scenarios).
productId	String	Product ID. This field is returned in the response when the node is yearly/monthly-billed with auto payment enabled.
maxPods	Integer	<p>Maximum number of pods that can be created on a node, including the default system pods. Value range: 16 to 256.</p> <p>This limit prevents the node from being overloaded of pods.</p> <p>The number of pods that can be created on a node is determined by multiple parameters. For details, see Maximum Number of Pods That Can Be Created on a Node.</p>
periodType	String	<ul style="list-style-type: none"> • month: The unit is month. • year: The unit is year. <p>NOTE</p> <p>Request parameter, which is valid and mandatory only when billingMode is set to 1 (yearly/monthly billing) or 2 (yearly/monthly billing with auto renewal enabled) (discarded).</p> <p>Response parameter, which is returned only when a yearly/monthly node is created.</p>

Parameter	Type	Description
periodNum	Integer	<p>Subscription duration. The value can be:</p> <ul style="list-style-type: none"> • If periodType is month, the value ranges from 1 to 9. • If periodType is year, the value is 1. <p>NOTE Request parameter, which is valid and mandatory when billingMode is set to 1 or 2 (discarded). Response parameter, which is returned only when a yearly/monthly node is created.</p>
isAutoRenew	String	<p>Whether auto renewal is enabled.</p> <ul style="list-style-type: none"> • true: Auto renewal is enabled. • false: Auto renewal is not enabled. <p>NOTE This field is valid when billingMode is set to 1 or 2 (discarded). If not specified, auto renewal is not enabled.</p>
isAutoPay	String	<p>Whether to deduct fees automatically.</p> <ul style="list-style-type: none"> • true: Enable automatic fee deduction. • false: Do not enable automatic fee deduction. <p>NOTE This field is valid when billingMode is set to 1 or 2 (discarded). If not specified when billingMode is set to 1, auto fee deduction is not enabled. (Discarded: If not specified when billingMode is set to 2, auto fee deduction is enabled.)</p>

Parameter	Type	Description
DockerLVMConfigOverride	String	<p>Docker data disk configuration item. (This parameter has been discarded. Use the storage field instead.) Example default configuration: "DockerLVMConfigOverride":"dockerThinpool=vgpaas/90%VG;kubernetesLV=vgpaas/10%VG;diskType=evs;lvType=linear"</p> <p>By default, if no VD disk is available, an error occurs because the data disk fails to be found. Set diskType based on the actual drive letter type.</p> <p>The following fields are included:</p> <ul style="list-style-type: none"> • userLV (optional): size of the user space, for example, vgpaas/20%VG. • userPath (optional): mount path of the user space, for example, /home/wqt-test. • diskType: disk type. Currently, only evs, hdd, and ssd are supported. • lvType: type of a logic volume. The value can be linear or striped. • dockerThinpool: Docker space size, for example, vgpaas/60%VG. • kubernetesLV: kubelet space size, for example, vgpaas/20%VG.
dockerBaseSize	Integer	<p>Available disk space of a single container on a node, in GB. This parameter has been discarded. Use containerBaseSize.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In Device Mapper mode, the default value is 10. In OverlayFS mode, the available space of a single container is not limited by default, and the dockerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters of the new versions.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that dockerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device.</p>

Parameter	Type	Description
containerBaseSize	Integer	<p>Available disk space of a single container on a node, in GB.</p> <p>If this parameter is left blank or is set to 0, the default value is used. In OverlayFS mode, the available space of a single container is not limited by default. In Device Mapper mode, the default value is 10, and the containerBaseSize setting takes effect only on nodes running EulerOS/Huawei Cloud EulerOS 2.0 in clusters v1.23.14-r0, v1.25.9-r0, v1.27.6-r0, v1.28.4-r0 or later.</p> <p>For details about how to allocate the space for the container runtime, see Data Disk Space Allocation.</p> <p>When Device Mapper is used, it is recommended that containerBaseSize be set to a value less than or equal to 80 GB. If the value is too large, the container runtime may fail to be started due to long initialization. If there are special requirements for the container disk space, you can mount an external or local storage device. In new versions, the Device Mapper mode is used only by BMSs in the same resource pool and is being discarded.</p>
publicKey	String	Public key of a node.
alpha.cce/preInstall	String	<p>Pre-installation script.</p> <p>NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)</p>
alpha.cce/postInstall	String	<p>Post-installation script.</p> <p>NOTE The input value must be Base64-encoded. (Command: echo -n "Content to be encoded" base64)</p>
alpha.cce/NodeImageID	String	This parameter is required when a custom image is used to create a BMS node.

Parameter	Type	Description
nicMultiqueue	String	<ul style="list-style-type: none"> Number of ENI queues. Example setting: <code>"[{"queue":4}]"</code> The following fields are included: <ul style="list-style-type: none"> queue: number of ENI queues. This field can be configured only for BMS nodes in a CCE Turbo cluster. Supported proportions are {"1":128, "2":92, "4":92, "8":32, "16":16, "28":9}. That is, if there is one queue, a maximum of 128 ENIs can be bound. If there are two queues, a maximum of 92 ENIs can be bound for two queues. A larger number of ENI queues indicates higher performance but fewer ENIs can be bound. The queue settings cannot be changed after the node pool is created.
nicThreshold	String	<ul style="list-style-type: none"> ENI pre-binding thresholds. Example setting: <code>"0.3:0.6"</code> <ul style="list-style-type: none"> Low threshold (L): determines the minimum number of pre-bound ENIs (Min). Formula: $Min = Total\ number\ of\ ENIs\ of\ the\ node \times L$ High threshold (H): determines the maximum number of pre-bound ENIs (Max). Formula: $Max = Total\ number\ of\ ENIs\ of\ the\ node \times H$ Number of ENIs bound to a BMS node (B) and number of ENIs being used by pods (U): $U + Min < B < U + Max$ If the number of pre-bound ENIs on a BMS node is smaller than the minimum allowed, the system will bind more ENIs to make the numbers equal. If the number of pre-bound ENIs on a BMS node is larger than the maximum allowed, the system periodically unbinds ENIs (about every 2 minutes) to make the numbers equal. Both the thresholds are one-decimal-place values ranging from 0.0 to 1.0. The low threshold must be smaller than or equal to the high one. This field can be configured only for BMS nodes in a CCE Turbo cluster. Pre-binding ENIs can speed up workload creation but occupies IP addresses.

Parameter	Type	Description
chargingMode	Integer	Billing mode of a node. This parameter has been deprecated. Use the billingMode parameter in NodeSpec .
agency_name	String	Name of an agency An agency is created by a tenant administrator on Identity and Access Management (IAM) to provide temporary credentials for CCE nodes to access cloud servers. This parameter is returned only when it is transferred during node creation.
kubeReservedMem	Integer	Reserved node memory, which is reserved for Kubernetes components.
systemReservedMem	Integer	Reserved node memory, which is reserved for system components.
init-node-password	String	Node password. If this parameter is used as a response parameter, asterisks (*) will be displayed.

Table 4-674 HostnameConfig

Parameter	Type	Description
type	String	Configuration type of the Kubernetes node name. The default value is privatelp . <ul style="list-style-type: none"> • privatelp: The Kubernetes node is named after its IP address. • cceNodeName: The Kubernetes node is named after the CCE node. <p>NOTE</p> <ul style="list-style-type: none"> • For a node which is configured using cceNodeName, the name is the same as the Kubernetes node name and the ECS name. The node name cannot be changed. If the ECS name is changed on the ECS console, the node name will retain unchanged after ECS synchronization. • For a node which is configured using cceNodeName, to avoid a conflict between Kubernetes nodes, the system automatically adds a suffix to each node name. The suffix is in the format of A hyphen (-) Five random characters. The value of the random characters is a lowercase letter or a digit ranging from 0 to 9.

Table 4-675 NodePoolNodeAutoscaling

Parameter	Type	Description
enable	Boolean	Whether to enable auto scaling.
minNodeCount	Integer	Minimum number of nodes allowed if auto scaling is enabled. The value cannot be greater than the maximum number of nodes allowed by the cluster specifications.
maxNodeCount	Integer	Maximum number of nodes allowed if auto scaling is enabled. This value must be greater than or equal to the value of minNodeCount and cannot exceed the maximum number of nodes in the cluster specifications.
scaleDownCooldownTime	Integer	Interval between two scaling operations, in minutes. During this period, nodes added after a scale-up will not be deleted.
priority	Integer	Weight of a node pool. A node pool with a higher weight has a higher priority during scaling.

Table 4-676 NodeManagement

Parameter	Type	Description
serverGroupReference	String	Cloud server group ID. If this field is specified, all nodes in the node pool will be created in this group. The group ID can be specified only when you create the node pool and cannot be modified. When you specify a cloud server group, the number of nodes in the node pool cannot exceed the group quota.

Table 4-677 SecurityID

Parameter	Type	Description
id	String	Security group ID

Table 4-678 DeleteNodePoolStatus

Parameter	Type	Description
currentNode	Integer	Total number of nodes in the current node pool (excluding the nodes that are being deleted).

Parameter	Type	Description
creatingNode	Integer	Number of nodes that are being created in the node pool.
deletingNode	Integer	Number of nodes that are being deleted in the current node pool.
phase	String	<p>Node pool status. Options:</p> <ul style="list-style-type: none"> • Null: The node pool is available. (The number of nodes in the current node pool has reached the expected value, and no node scaling is being performed.) • Synchronizing: The node pool is being scaled. (The number of nodes in the current node pool has not reached the expected value, and no node scaling is being performed.) • Synchronized: The node pool scaling is pending. (The number of nodes in the current node pool has not reached the expected value, or node scaling is being performed.) • SoldOut: The node pool cannot be scaled out. (This field is used in multiple scenarios, for example, resources in the node pool have been sold out or the resource quota is insufficient.) <p>NOTE This node pool status has been deprecated. It is reserved only for compatibility. Do not use it anymore. Use the following instead:</p> <ul style="list-style-type: none"> • Node pool scaling status: You can obtain the status of the current node pool using parameters such as currentNode, creatingNode, or deletingNode. • Node pool scale-out: You can use conditions to obtain the detailed status of a node pool. Scalable can replace SoldOut. • Deleting: The node pool is being deleted. • Error: An error occurred in the node pool.
jobId	String	ID of a job executed on the node pool.
conditions	Array of NodePoolCondition objects	Node pool status details. For details, see the definition of Condition .

Table 4-679 NodePoolCondition

Parameter	Type	Description
type	String	Condition type. The options are as follows: <ul style="list-style-type: none"> • Scalable: whether a node pool can be scaled. If the status is False, node pool scaling will not be triggered again. • QuotaInsufficient: Quotas on which node pool scaling depends are insufficient, affecting the node pool scaling status. • ResourceInsufficient: Resources on which node pool scaling depends are insufficient, affecting the node pool scaling status. • UnexpectedError: The node pool fails to be scaled out due to unexpected reasons, affecting the node pool scaling status. • LockedByOrder: The yearly/monthly-billed node pool is locked due to an order. The value of Reason is the ID of the pending order. • Error: A node pool error occurs. A common trigger is deletion failure.
status	String	Current status of Condition . The options are as follows: <ul style="list-style-type: none"> • "True" • "False"
lastProbeTime	String	Time when the status was last checked
lastTransitTime	String	Time when the status was last changed
reason	String	Reason why the status was last changed
message	String	Detailed condition description

Example Requests

None

Example Responses

Status code: 200

The job for deleting the node pool is successfully delivered.

```
{
  "kind": "NodePool",
  "apiVersion": "v3",
  "metadata": {
    "name": "lc-it-nodepool-79796",
```

```
"uid" : "99addaa2-69eb-11ea-a592-0255ac1001bb"
},
"spec" : {
  "type" : "vm",
  "nodeTemplate" : {
    "flavor" : "s6.large.2",
    "az" : "*****",
    "os" : "EulerOS 2.5",
    "login" : {
      "sshKey" : "KeyPair-001"
    }
  },
  "rootVolume" : {
    "volumetype" : "SAS",
    "size" : 40
  },
  "dataVolumes" : [ {
    "volumetype" : "SAS",
    "size" : 100,
    "extendParam" : {
      "useType" : "docker"
    }
  } ],
  "publicIP" : {
    "eip" : {
      "bandwidth" : { }
    }
  },
  "nodeNicSpec" : {
    "primaryNic" : {
      "subnetId" : "7e767d10-7548-4df5-ad72-aeac1d08bd8a"
    }
  },
  "billingMode" : 0,
  "extendParam" : {
    "maxPods" : 110
  },
  "k8sTags" : {
    "cce.cloud.com/cce-nodepool" : "lc-it-nodepool-79796"
  }
},
"autoscaling" : { },
"nodeManagement" : { }
},
"status" : {
  "phase" : "Deleting",
  "jobId" : "3281fa02-69ee-11ea-a592-0255ac1001bb"
}
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class DeleteNodePoolSolution {
```

```
public static void main(String[] args) {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running
    // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    String ak = System.getenv("CLOUD_SDK_AK");
    String sk = System.getenv("CLOUD_SDK_SK");
    String projectId = "{project_id}";

    ICredential auth = new BasicCredentials()
        .withProjectId(projectId)
        .withAk(ak)
        .withSk(sk);

    CceClient client = CceClient.newBuilder()
        .withCredential(auth)
        .withRegion(CceRegion.valueOf("<YOUR REGION>"))
        .build();
    DeleteNodePoolRequest request = new DeleteNodePoolRequest();
    request.withClusterId("{cluster_id}");
    request.withNodepoolId("{nodepool_id}");
    try {
        DeleteNodePoolResponse response = client.deleteNodePool(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteNodePoolRequest()
        request.cluster_id = "{cluster_id}"
```

```
request.nodepool_id = "{nodepool_id}"
response = client.delete_node_pool(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteNodePoolRequest{}
    request.ClusterId = "{cluster_id}"
    request.NodepoolId = "{nodepool_id}"
    response, err := client.DeleteNodePool(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The job for deleting the node pool is successfully delivered.

Error Codes

See [Error Codes](#).

4.5 Storage Management

4.5.1 Creating a PVC (to be discarded)

Function

This API is used to create a PersistentVolumeClaim (PVC) in a specified namespace. A PVC is a request for PersistentVolume (PV). The PV carries details of real cloud storage such as EVS disks, SFS file systems, and OBS buckets. This API is to be deprecated. Use the corresponding Kubernetes PVC API instead.

 **NOTE**

The URL for storage management is in the format of **https://{clusterid}.Endpoint/uri**. In the URL, *{clusterid}* indicates the cluster ID, and *uri* indicates the resource path, that is, the path for API access. If *https://Endpoint/uri* is used, the **X-Cluster-ID** parameter in the request header must be specified.

Calling Method

For details, see [Calling APIs](#).

URI

POST /api/v1/namespaces/{namespace}/cloudpersistentvolumeclaims

Table 4-680 Path Parameters

Parameter	Mandatory	Type	Description
namespace	Yes	String	Namespace where the PVC is located. Constraints: <ul style="list-style-type: none"> • Custom namespace. You must create a namespace in the cluster before using it. • The namespace provided by the system is namespace default. • Namespaces kube-system and kube-public cannot be used.

Request Parameters

Table 4-681 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .
X-Cluster-ID	No	String	Cluster ID. This field is mandatory when the URL format https://Endpoint/uri is used. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Table 4-682 Request body parameters

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	API version. The value is fixed at v1 .
kind	Yes	String	API type. The value is fixed at PersistentVolumeClaim .
metadata	Yes	PersistentVolumeClaimMetadata object	Cluster object metadata, which is a collection of attributes.
spec	Yes	PersistentVolumeClaimSpec object	Detailed description of the cluster object. CCE creates or updates objects by defining or updating spec .
status	No	PersistentVolumeClaimStatus object	PVC status. The request to create a PVC does not need to carry this parameter.

Table 4-683 PersistentVolumeClaimMetadata

Parameter	Mandatory	Type	Description
name	Yes	String	PVC name. Enter 1 to 253 characters starting and ending with a letter or digit. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed. PVC names must be unique in a namespace.
labels	No	String	PVC labels, in the format of key-value pairs. <ul style="list-style-type: none"> • Key: Enter 1 to 63 characters, starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain can be prefixed to a key and contain a maximum of 253 characters. Example DNS subdomain: example.com/my-key • Value: The value can be left blank or contain 1 to 63 characters that start with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed in the character string.

Table 4-684 PersistentVolumeClaimSpec

Parameter	Mandatory	Type	Description
volumeID	Yes	String	ID of an existing storage volume. <ul style="list-style-type: none"> • If an SFS, EVS, or SFS Turbo volume is used, set this parameter to the ID of the volume. • If an OBS bucket is used, set this parameter to the OBS bucket name.

Parameter	Mandatory	Type	Description
storageType	Yes	String	Cloud storage class. This parameter is used together with volumeID . That is, volumeID and storageType must be configured at the same time. <ul style="list-style-type: none"> • bs: EVS • nfs: SFS • obs: OBS • efs: SFS Turbo
accessModes	Yes	Array of strings	Access mode of the volume. Only the first value in all selected options is valid. <ul style="list-style-type: none"> • ReadWriteOnce: The volume can be mounted as read-write by a single node. <p>NOTE This function is supported only when the cluster version is v1.13.10 and the storage-driver version is 1.0.19.</p> <ul style="list-style-type: none"> • ReadOnlyMany (default): The volume can be mounted as read-only by many nodes. • ReadWriteMany: The volume can be mounted as read-write by many nodes.
storageClassName	No	String	Storage class name of the PVC.
volumeName	No	String	Name of the PV bound to the PVC.
resources	No	ResourceRequirements object	Resource requests and limits.
volumeMode	No	String	PV type specified by the PVC.

Table 4-685 ResourceRequirements

Parameter	Mandatory	Type	Description
limits	No	Map<String,String>	Resource limits. This parameter is invalid during creation.

Parameter	Mandatory	Type	Description
requests	No	Map<String,String>	Resource requests. This parameter is invalid during creation.

Table 4-686 PersistentVolumeClaimStatus

Parameter	Mandatory	Type	Description
accessModes	No	Array of strings	Access mode of the PV.
capacity	No	String	Actual resources and capacity of the PV.
phase	No	String	Current status of the PVC.

Response Parameters

Status code: 201

Table 4-687 Response body parameters

Parameter	Type	Description
apiVersion	String	API version. The value is fixed at v1 .
kind	String	API type. The value is fixed at PersistentVolumeClaim .
metadata	PersistentVolumeClaimMetadata object	Cluster object metadata, which is a collection of attributes.
spec	PersistentVolumeClaimSpec object	Detailed description of the cluster object. CCE creates or updates objects by defining or updating spec .
status	PersistentVolumeClaimStatus object	PVC status. The request to create a PVC does not need to carry this parameter.

Table 4-688 PersistentVolumeClaimMetadata

Parameter	Type	Description
name	String	PVC name. Enter 1 to 253 characters starting and ending with a letter or digit. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed. PVC names must be unique in a namespace.
labels	String	PVC labels, in the format of key-value pairs. <ul style="list-style-type: none"> • Key: Enter 1 to 63 characters, starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain can be prefixed to a key and contain a maximum of 253 characters. Example DNS subdomain: example.com/my-key • Value: The value can be left blank or contain 1 to 63 characters that start with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed in the character string.

Table 4-689 PersistentVolumeClaimSpec

Parameter	Type	Description
volumeID	String	ID of an existing storage volume. <ul style="list-style-type: none"> • If an SFS, EVS, or SFS Turbo volume is used, set this parameter to the ID of the volume. • If an OBS bucket is used, set this parameter to the OBS bucket name.
storageType	String	Cloud storage class. This parameter is used together with volumeID . That is, volumeID and storageType must be configured at the same time. <ul style="list-style-type: none"> • bs: EVS • nfs: SFS • obs: OBS • efs: SFS Turbo

Parameter	Type	Description
accessModes	Array of strings	Access mode of the volume. Only the first value in all selected options is valid. <ul style="list-style-type: none"> • ReadWriteOnce: The volume can be mounted as read-write by a single node. <p>NOTE This function is supported only when the cluster version is v1.13.10 and the storage-driver version is 1.0.19.</p> <ul style="list-style-type: none"> • ReadOnlyMany (default): The volume can be mounted as read-only by many nodes. • ReadWriteMany: The volume can be mounted as read-write by many nodes.
storageClassName	String	Storage class name of the PVC.
volumeName	String	Name of the PV bound to the PVC.
resources	ResourceRequirements object	Resource requests and limits.
volumeMode	String	PV type specified by the PVC.

Table 4-690 ResourceRequirements

Parameter	Type	Description
limits	Map<String,String>	Resource limits. This parameter is invalid during creation.
requests	Map<String,String>	Resource requests. This parameter is invalid during creation.

Table 4-691 PersistentVolumeClaimStatus

Parameter	Type	Description
accessModes	Array of strings	Access mode of the PV.
capacity	String	Actual resources and capacity of the PV.
phase	String	Current status of the PVC.

Example Requests

Specifying an EVS volume ID and creating a PVC

```
POST /api/v1/namespaces/default/cloudpersistentvolumeclaims

{
  "apiVersion": "v1",
  "kind": "PersistentVolumeClaim",
  "metadata": {
    "name": "csms-dev-create",
    "namespace": "default"
  },
  "spec": {
    "volumeID": "86b29e16-23db-11e7-9c83-fa163ec08232",
    "storageType": "bs",
    "accessModes": [ "ReadWriteMany" ]
  }
}
```

Example Responses

Status code: 201

The job for creating a PVC is successfully delivered.

```
{
  "kind": "PersistentVolumeClaim",
  "apiVersion": "v1",
  "metadata": {
    "name": " csms-dev-create ",
    "namespace": "default",
    "selfLink": "/api/v1/namespaces/default/persistentvolumeclaims/db-mysql-0",
    "uid": "86b29e16-23db-11e7-9c83-fa163ec08232",
    "resourceVersion": "1793115",
    "creationTimestamp": "2017-04-18T02:05:42Z"
  },
  "spec": {
    "volumeName": "csms-dev-create ",
    "accessModes": [ "ReadWriteMany" ],
    "resources": {
      "requests": {
        "storage": "1Gi"
      }
    }
  },
  "status": {
    "phase": "Pending",
    "accessModes": [ "ReadWriteMany" ]
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Specifying an EVS volume ID and creating a PVC

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;
```

```
import java.util.List;
import java.util.ArrayList;

public class CreateCloudPersistentVolumeClaimsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();

        CreateCloudPersistentVolumeClaimsRequest request = new
        CreateCloudPersistentVolumeClaimsRequest();
        request.withNamespace("{namespace}");
        PersistentVolumeClaim body = new PersistentVolumeClaim();
        List<PersistentVolumeClaimSpec.AccessModesEnum> listSpecAccessModes = new ArrayList<>();

        listSpecAccessModes.add(PersistentVolumeClaimSpec.AccessModesEnum.fromValue("ReadWriteMany"));
        PersistentVolumeClaimSpec specbody = new PersistentVolumeClaimSpec();
        specbody.withVolumeID("86b29e16-23db-11e7-9c83-fa163ec08232")
            .withStorageType("bs")
            .withAccessModes(listSpecAccessModes);
        PersistentVolumeClaimMetadata metadatabody = new PersistentVolumeClaimMetadata();
        metadatabody.withName("csms-dev-create");
        body.withSpec(specbody);
        body.withMetadata(metadatabody);
        body.withKind("PersistentVolumeClaim");
        body.withApiVersion("v1");
        request.withBody(body);
        try {
            CreateCloudPersistentVolumeClaimsResponse response =
            client.createCloudPersistentVolumeClaims(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Specifying an EVS volume ID and creating a PVC

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *
```

```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateCloudPersistentVolumeClaimsRequest()
        request.namespace = "{namespace}"
        listAccessModesSpec = [
            "ReadWriteMany"
        ]
        specbody = PersistentVolumeClaimSpec(
            volume_id="86b29e16-23db-11e7-9c83-fa163ec08232",
            storage_type="bs",
            access_modes=listAccessModesSpec
        )
        metadatabody = PersistentVolumeClaimMetadata(
            name="csms-dev-create"
        )
        request.body = PersistentVolumeClaim(
            spec=specbody,
            metadata=metadatabody,
            kind="PersistentVolumeClaim",
            api_version="v1"
        )
        response = client.create_cloud_persistent_volume_claims(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Specifying an EVS volume ID and creating a PVC

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
```

```

WithAk(ak).
WithSk(sk).
Build()

client := cce.NewCceClient(
    cce.CceClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateCloudPersistentVolumeClaimsRequest{}
request.Namespace = "{namespace}"
var listAccessModesSpec = []model.PersistentVolumeClaimSpecAccessModes{
    model.GetPersistentVolumeClaimSpecAccessModesEnum().READ_WRITE_MANY,
}
specbody := &model.PersistentVolumeClaimSpec{
    VolumeID: "86b29e16-23db-11e7-9c83-fa163ec08232",
    StorageType: "bs",
    AccessModes: listAccessModesSpec,
}
metadatabody := &model.PersistentVolumeClaimMetadata{
    Name: "csms-dev-create",
}
request.Body = &model.PersistentVolumeClaim{
    Spec: specbody,
    Metadata: metadatabody,
    Kind: "PersistentVolumeClaim",
    ApiVersion: "v1",
}
response, err := client.CreateCloudPersistentVolumeClaims(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
201	The job for creating a PVC is successfully delivered.

Error Codes

See [Error Codes](#).

4.5.2 Deleting a PVC (to be discarded)

Function

This API is used to delete a PVC from a specified namespace. Using this API, you can determine whether to retain the cloud storage volume associated with the

PVC. This API has been deprecated. Use the corresponding Kubernetes PV API instead.

 NOTE

The URL for storage management is in the format of **https://{clusterid}.Endpoint/uri**. In the URL, *{clusterid}* indicates the cluster ID, and *uri* indicates the resource path, that is, the path for API access. If *https://Endpoint/uri* is used, the **X-Cluster-ID** parameter in the request header must be specified.

Calling Method

For details, see [Calling APIs](#).

URI

DELETE /api/v1/namespaces/{namespace}/cloudpersistentvolumeclaims/{name}

Table 4-692 Path Parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the PVC to be deleted.
namespace	Yes	String	Namespace where the PVC is located.

Table 4-693 Query Parameters

Parameter	Mandatory	Type	Description
deleteVolume	No	String	Whether to delete associated backend cloud storage volume after the PVC is deleted. The value can be true (delete) or false (not delete). The value defaults to false .
storageType	No	String	Cloud storage type. This parameter is used together with deleteVolume. That is, deleteVolume and storageType must be configured at the same time. <ul style="list-style-type: none"> ● bs: EVS ● nfs: SFS ● obs: OBS ● efs: SFS Turbo

Request Parameters

Table 4-694 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .
X-Cluster-ID	No	String	Cluster ID. This field is mandatory when the URL format https://Endpoint/uri is used. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Response Parameters

Status code: 200

Table 4-695 Response body parameters

Parameter	Type	Description
apiVersion	String	API version. The value is fixed at v1 .
kind	String	API type. The value is fixed at PersistentVolumeClaim .
metadata	PersistentVolumeClaimMetadata object	Cluster object metadata, which is a collection of attributes.
spec	PersistentVolumeClaimSpec object	Detailed description of the cluster object. CCE creates or updates objects by defining or updating spec .
status	PersistentVolumeClaimStatus object	PVC status. The request to create a PVC does not need to carry this parameter.

Table 4-696 PersistentVolumeClaimMetadata

Parameter	Type	Description
name	String	PVC name. Enter 1 to 253 characters starting and ending with a letter or digit. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed. PVC names must be unique in a namespace.
labels	String	PVC labels, in the format of key-value pairs. <ul style="list-style-type: none"> • Key: Enter 1 to 63 characters, starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain can be prefixed to a key and contain a maximum of 253 characters. Example DNS subdomain: example.com/my-key • Value: The value can be left blank or contain 1 to 63 characters that start with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed in the character string.

Table 4-697 PersistentVolumeClaimSpec

Parameter	Type	Description
volumeID	String	ID of an existing storage volume. <ul style="list-style-type: none"> • If an SFS, EVS, or SFS Turbo volume is used, set this parameter to the ID of the volume. • If an OBS bucket is used, set this parameter to the OBS bucket name.
storageType	String	Cloud storage class. This parameter is used together with volumeID . That is, volumeID and storageType must be configured at the same time. <ul style="list-style-type: none"> • bs: EVS • nfs: SFS • obs: OBS • efs: SFS Turbo

Parameter	Type	Description
accessModes	Array of strings	Access mode of the volume. Only the first value in all selected options is valid. <ul style="list-style-type: none"> • ReadWriteOnce: The volume can be mounted as read-write by a single node. <p>NOTE This function is supported only when the cluster version is v1.13.10 and the storage-driver version is 1.0.19.</p> <ul style="list-style-type: none"> • ReadOnlyMany (default): The volume can be mounted as read-only by many nodes. • ReadWriteMany: The volume can be mounted as read-write by many nodes.
storageClassName	String	Storage class name of the PVC.
volumeName	String	Name of the PV bound to the PVC.
resources	ResourceRequirements object	Resource requests and limits.
volumeMode	String	PV type specified by the PVC.

Table 4-698 ResourceRequirements

Parameter	Type	Description
limits	Map<String,String>	Resource limits. This parameter is invalid during creation.
requests	Map<String,String>	Resource requests. This parameter is invalid during creation.

Table 4-699 PersistentVolumeClaimStatus

Parameter	Type	Description
accessModes	Array of strings	Access mode of the PV.
capacity	String	Actual resources and capacity of the PV.
phase	String	Current status of the PVC.

Example Requests

None

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class DeleteCloudPersistentVolumeClaimsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();

        DeleteCloudPersistentVolumeClaimsRequest request = new
        DeleteCloudPersistentVolumeClaimsRequest();
        request.setName("{name}");
        request.withNamespace("{namespace}");
        try {
            DeleteCloudPersistentVolumeClaimsResponse response =
            client.deleteCloudPersistentVolumeClaims(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteCloudPersistentVolumeClaimsRequest()
        request.name = "{name}"
        request.namespace = "{namespace}"
        response = client.delete_cloud_persistent_volume_claims(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteCloudPersistentVolumeClaimsRequest{}
    request.Name = "{name}"
    request.Namespace = "{namespace}"
    response, err := client.DeleteCloudPersistentVolumeClaims(request)
```

```

if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The job for deleting a PVC is successfully delivered.

Error Codes

See [Error Codes](#).

4.6 Add-on Management

4.6.1 Installing an Add-on Instance

Function

This API is used to install an add-on instance by using an add-on template.

Calling Method

For details, see [Calling APIs](#).

URI

POST /api/v3/addons

Request Parameters

Table 4-700 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Table 4-701 Request body parameters

Parameter	Mandatory	Type	Description
kind	Yes	String	API type. The value is fixed at Addon and cannot be changed. Any user-defined value is invalid.
apiVersion	Yes	String	API version. The value is fixed at v3 and cannot be changed. Any user-defined value is invalid.
metadata	Yes	AddonMetadata object	Basic information about the object. Metadata is a collection of attributes.
spec	Yes	InstanceRequestSpec object	Detailed description of add-on installation or upgrade.

Table 4-702 AddonMetadata

Parameter	Mandatory	Type	Description
uid	No	String	Unique ID
name	No	String	Add-on name
alias	No	String	Add-on alias
labels	No	Map<String,String>	Add-on labels in key-value pairs. This is a reserved field and does not take effect.

Parameter	Mandatory	Type	Description
annotations	No	Map<String,String>	Add-on annotations in the format of key-value pairs. <ul style="list-style-type: none"> For add-on installation, the value is fixed at {"addon.install/type":"install"}. For add-on upgrade, the value is fixed at {"addon.upgrade/type":"upgrade"}.
updateTimestamp	No	String	Updated at
creationTimestamp	No	String	Created at

Table 4-703 InstanceRequestSpec

Parameter	Mandatory	Type	Description
version	No	String	Version of the add-on to install or upgrade, for example, 1.0.0 . <ul style="list-style-type: none"> Installation: This parameter is optional. If not specified, the latest version supported by the cluster is used. Upgrade: This parameter is mandatory. The version number must be specified.
clusterID	Yes	String	Cluster ID.
values	Yes	Map<String,Object>	Add-on template installation parameters (varying depending on the add-on). During the add-on upgrade, you need to specify all the installation parameters. If the parameters are not specified, the default values in the add-on template are used. The current add-on installation parameters can be obtained through the API for querying add-on instances.

Parameter	Mandatory	Type	Description
addonTemplate	Yes	String	Name of the add-on template to be installed, for example, coredns .

Response Parameters

Status code: 201

Table 4-704 Response body parameters

Parameter	Type	Description
kind	String	API type. The value is fixed at Addon and cannot be changed.
apiVersion	String	API version. The value is fixed at v3 and cannot be changed.
metadata	AddonMetadata object	Basic information about the object. Metadata is a collection of attributes.
spec	InstanceSpec object	Detailed description of the add-on instance.
status	AddonInstanceStatus object	Add-on instance status.

Table 4-705 AddonMetadata

Parameter	Type	Description
uid	String	Unique ID
name	String	Add-on name
alias	String	Add-on alias
labels	Map<String,String>	Add-on labels in key-value pairs. This is a reserved field and does not take effect.
annotations	Map<String,String>	Add-on annotations in the format of key-value pairs. <ul style="list-style-type: none"> For add-on installation, the value is fixed at {"addon.install/type":"install"}. For add-on upgrade, the value is fixed at {"addon.upgrade/type":"upgrade"}.

Parameter	Type	Description
updateTimestamp	String	Updated at
creationTimestamp	String	Created at

Table 4-706 InstanceSpec

Parameter	Type	Description
clusterID	String	Cluster ID.
version	String	Add-on template version, for example, 1.0.0 .
addonTemplateName	String	Add-on template name, for example, coredns .
addonTemplateType	String	Add-on template type.
addonTemplateLogo	String	URL for obtaining the add-on template logo.
addonTemplateLabels	Array of strings	Labels of the add-on template.
description	String	Add-on template description.
values	Map<String,Object>	Add-on template installation parameters (varying depending on the add-on). Set the parameters accordingly.

Table 4-707 AddonInstanceStatus

Parameter	Type	Description
status	String	<p>Add-on instance status. Options:</p> <ul style="list-style-type: none"> ● running: All of the add-on instances are running. This specifies that the add-on runs properly. ● abnormal: The add-on instances are abnormal and the add-on cannot be used. You can click the add-on name to view exceptions. ● installing: The add-on is being installed. ● installFailed: Installing the add-on failed. In this case, uninstall the add-on and then reinstall it. ● upgrading: The add-on is being upgraded. ● upgradeFailed: Upgrading the add-on failed. In this case, upgrade the add-on again or uninstall the add-on and reinstall it. ● deleting: The add-on is being deleted. ● deleteFailed: Deleting the add-on failed. In this case, uninstall the add-on again. ● deleteSuccess: Deleting the add-on succeeded. ● available: Only some instances of the add-on are running. This specifies that some functions of the add-on are available. ● rollbacking: The add-on is being rolled back. ● rollbackFailed: Rolling back the add-on failed. In this case, roll back the add-on again or uninstall the add-on and reinstall it. ● unknown: The add-on chart instance does not exist.
Reason	String	Cause of the add-on installation failure.
message	String	Installation error details.
targetVersions	Array of strings	Versions to which the current add-on version can be upgraded.
currentVersion	Versions object	Information about the current add-on version.
isRollbackable	Boolean	Whether the add-on version can be rolled back to the source version.

Parameter	Type	Description
previousVersion	String	The add-on version before upgrade or rollback

Table 4-708 Versions

Parameter	Type	Description
version	String	Add-on version.
input	Object	Add-on installation parameters.
stable	Boolean	Whether the add-on version is a stable release.
translate	Object	Translation information used by the GUI.
supportVersions	Array of SupportVersions objects	Cluster versions that support the add-on.
creationTimestamp	String	Creation time.
updateTimestamp	String	Update time.

Table 4-709 SupportVersions

Parameter	Type	Description
clusterType	String	Cluster type that supports the add-on.
clusterVersion	Array of strings	Cluster versions that support the add-on. The value is a regular expression.

Example Requests

Install the coredns add-on of version 1.17.15. The add-on specification is 2500 QPS, and the number of add-on instances is 2.

```
{
  "kind": "Addon",
  "apiVersion": "v3",
  "metadata": {
    "annotations": {
      "addon.install/type": "install"
    }
  },
  "spec": {
    "clusterID": "1b2ec02d-a3b2-11ec-b0d0-0255ac100099",
    "version": "1.17.15",
    "addonTemplateName": "coredns",
    "values": {
```

```

"basic" : {
  "cluster_ip" : "10.247.3.10",
  "image_version" : "1.17.15",
  "platform" : "linux-amd64",
  "swr_addr" : "<Replace_SWR_address>",
  "swr_user" : "hwofficial",
  "rbac_enabled" : true
},
"flavor" : {
  "name" : 2500,
  "replicas" : 2,
  "resources" : [ {
    "limitsCpu" : "500m",
    "limitsMem" : "512Mi",
    "name" : "coredns",
    "requestsCpu" : "500m",
    "requestsMem" : "512Mi"
  } ]
},
"custom" : {
  "stub_domains" : { },
  "upstream_nameservers" : [ ],
  "cluster_id" : "1b2ec02d-a3b2-11ec-b0d0-0255ac100099",
  "tenant_id" : "0504201b6c80256b2f08c0099f0c8fe4"
}
}
}
}

```

Example Responses

Status code: 201

OK

```

{
  "kind" : "Addon",
  "apiVersion" : "v3",
  "metadata" : {
    "uid" : "b748aaea-a984-11ec-987b-0255ac1000bc",
    "name" : "coredns",
    "alias" : "coredns",
    "creationTimestamp" : "2022-03-22T02:06:41Z",
    "updateTimestamp" : "2022-03-22T02:06:41Z"
  },
  "spec" : {
    "clusterID" : "1b2ec02d-a3b2-11ec-b0d0-0255ac100099",
    "version" : "1.17.15",
    "addonTemplateName" : "coredns",
    "addonTemplateType" : "helm",
    "addonTemplateLogo" : "",
    "addonTemplateLabels" : [ "ServiceDiscovery" ],
    "description" : "CoreDNS is a DNS server that chains plugins and provides Kubernetes DNS Services",
    "values" : {
      "basic" : {
        "cluster_ip" : "10.247.3.10",
        "image_version" : "1.17.15",
        "platform" : "linux-amd64",
        "rbac_enabled" : true,
        "swr_addr" : "",
        "swr_user" : "hwofficial"
      },
      "custom" : {
        "cluster_id" : "1b2ec02d-a3b2-11ec-b0d0-0255ac100099",
        "stub_domains" : { },
        "tenant_id" : "0504201b6c80256b2f08c0099f0c8fe4",
        "upstream_nameservers" : [ ]
      },
      "flavor" : {

```

```

"name" : 2500,
"replicas" : 2,
"resources" : [ {
  "limitsCpu" : "500m",
  "limitsMem" : "512Mi",
  "name" : "coredns",
  "requestsCpu" : "500m",
  "requestsMem" : "512Mi"
} ]
}
},
"status" : {
  "status" : "installing",
  "Reason" : "",
  "message" : "",
  "targetVersions" : null,
  "currentVersion" : {
    "version" : "1.17.15",
    "input" : {
      "basic" : {
        "cluster_ip" : "10.247.3.10",
        "image_version" : "1.17.15",
        "platform" : "linux-amd64",
        "swr_addr" : "",
        "swr_user" : "hwofficial"
      },
      "parameters" : {
        "custom" : {
          "stub_domains" : "",
          "upstream_nameservers" : ""
        },
        "flavor1" : {
          "name" : 2500,
          "replicas" : 2,
          "resources" : [ {
            "limitsCpu" : "500m",
            "limitsMem" : "512Mi",
            "name" : "coredns",
            "requestsCpu" : "500m",
            "requestsMem" : "512Mi"
          } ]
        },
        "flavor2" : {
          "name" : 5000,
          "replicas" : 2,
          "resources" : [ {
            "limitsCpu" : "1000m",
            "limitsMem" : "1024Mi",
            "name" : "coredns",
            "requestsCpu" : "1000m",
            "requestsMem" : "1024Mi"
          } ]
        },
        "flavor3" : {
          "name" : 10000,
          "replicas" : 2,
          "resources" : [ {
            "limitsCpu" : "2000m",
            "limitsMem" : "2048Mi",
            "name" : "coredns",
            "requestsCpu" : "2000m",
            "requestsMem" : "2048Mi"
          } ]
        },
        "flavor4" : {
          "name" : 20000,
          "replicas" : 4,
          "resources" : [ {

```

```

        "limitsCpu" : "2000m",
        "limitsMem" : "2048Mi",
        "name" : "coredns",
        "requestsCpu" : "2000m",
        "requestsMem" : "2048Mi"
    } ]
}
},
"stable" : true,
"translate" : {
    "en_US" : {
        "addon" : {
            "changeLog" : "Supported CCE clusters of v1.21.",
            "description" : "CoreDNS is a DNS server that chains plugins and provides Kubernetes DNS Services"
        },
        "description" : {
            "Parameters.custom.stub_domains" : "The target nameserver may itself be a Kubernetes service. For instance, you can run your own copy of dnsmasq to export custom DNS names into the ClusterDNS namespace, a JSON map using a DNS suffix key (e.g. \"acme.local\") and a value consisting of a JSON array of DNS IPs.",
            "Parameters.custom.upstream_nameservers" : "If specified, then the values specified replace the nameservers taken by default from the node's /etc/resolv.conf. Limits:a maximum of three upstream nameservers can be specified, A JSON array of DNS IPs.",
            "Parameters.flavor1.description" : "Concurrent domain name resolution ability - External domain name: 2500 qps, Internal domain name: 10000 qps",
            "Parameters.flavor1.name" : 2500,
            "Parameters.flavor2.description" : "Concurrent domain name resolution ability - External domain name: 5000 qps, Internal domain name: 20000 qps",
            "Parameters.flavor2.name" : 5000,
            "Parameters.flavor3.description" : "Concurrent domain name resolution ability - External domain name: 10000 qps, Internal domain name: 40000 qps",
            "Parameters.flavor3.name" : 10000,
            "Parameters.flavor4.description" : "Concurrent domain name resolution ability - External domain name: 20000 qps, Internal domain name: 80000 qps",
            "Parameters.flavor4.name" : 20000
        },
        "key" : {
            "Parameters.custom.stub_domains" : "stub domain",
            "Parameters.custom.upstream_nameservers" : "upstream nameservers"
        }
    },
    "fr_FR" : {
        "addon" : {
            "changeLog" : "Prise en charge du cluster 1.21.",
            "description" : "Un serveur DNS qui enchaîne les plug-ins et fournit des services DNS Kubernetes."
        },
        "description" : {
            "Parameters.custom.stub_domains" : "Le serveur de noms cible peut lui-même être un service Kubernetes. Par exemple, vous pouvez exécuter votre propre copie de dnsmasq pour exporter des noms DNS personnalisés dans l'espace de noms ClusterDNS, une carte JSON à l'aide d'une clé de suffixe DNS (par exemple, «acme.local») et une valeur constituée d'un tableau JSON d'adresses IP DNS.",
            "Parameters.custom.upstream_nameservers" : "Si spécifié, les valeurs spécifiées remplacent les serveurs de noms pris par défaut dans le fichier /etc/resolv.conf du nœud. Limites: un maximum de trois serveurs de noms en amont peuvent être spécifiés, un tableau JSON d'adresses IP DNS.",
            "Parameters.flavor1.description" : "Capacité de résolution de nom de domaine simultanée - Nom de domaine externe: 2500 qps, Nom de domaine interne: 10000 qp",
            "Parameters.flavor1.name" : 2500,
            "Parameters.flavor2.description" : "Capacité de résolution de nom de domaine simultanée - Nom de domaine externe: 5000 qps, Nom de domaine interne: 20000 qp",
            "Parameters.flavor2.name" : 5000,
            "Parameters.flavor3.description" : "Capacité de résolution de nom de domaine simultanée - Nom de domaine externe: 10000 qps, Nom de domaine interne: 40000 qp",
            "Parameters.flavor3.name" : 10000,
            "Parameters.flavor4.description" : "Capacité de résolution de nom de domaine simultanée - Nom de domaine externe: 20000 qps, Nom de domaine interne: 80000 qp",
            "Parameters.flavor4.name" : 20000
        },
        "key" : {

```



```
    "Parameters.custom.stub_domains" : "domaine stub",
    "Parameters.custom.upstream_nameservers" : "serveurs de noms en amont"
  }
},
"zh_CN" : {
  "addon" : {
    "changeLog" : "",
    "description" : ""
  },
  "description" : {
    "Parameters.custom.stub_domains" : "",
    "Parameters.custom.upstream_nameservers" : "",
    "Parameters.flavor1.description" : "",
    "Parameters.flavor1.name" : 2500,
    "Parameters.flavor2.description" : "",
    "Parameters.flavor2.name" : 5000,
    "Parameters.flavor3.description" : "",
    "Parameters.flavor3.name" : 10000,
    "Parameters.flavor4.description" : "",
    "Parameters.flavor4.name" : 20000
  },
  "key" : {
    "Parameters.custom.stub_domains" : "",
    "Parameters.custom.upstream_nameservers" : ""
  }
}
},
"supportVersions" : null,
"creationTimestamp" : "2021-12-14T13:43:15Z",
"updateTimestamp" : "2022-01-11T14:32:10Z"
}
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Install the coredns add-on of version 1.17.15. The add-on specification is 2500 QPS, and the number of add-on instances is 2.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

import java.util.Map;
import java.util.HashMap;

public class CreateAddonInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
    }
}
```

```
ICredential auth = new BasicCredentials()
    .withAk(ak)
    .withSk(sk);

CceClient client = CceClient.newBuilder()
    .withCredential(auth)
    .withRegion(CceRegion.valueOf("<YOUR REGION>"))
    .build();
CreateAddonInstanceRequest request = new CreateAddonInstanceRequest();
InstanceRequest body = new InstanceRequest();
Map<String, Object> listSpecValues = new HashMap<>();
listSpecValues.put("basic", "{\rbac_enabled\":true,\swr_user\":\hwofficial\",image_version
\":\1.17.15\",cluster_ip\":\10.247.3.10\",platform\":\linux-amd64\",swr_addr
\":\<Replace_SWR_address>\");
listSpecValues.put("flavor", "{\replicas\":2,\name\":2500,\resources\":[{\limitsCpu\":\500m
\",name\":\coredns\",limitsMem\":\512Mi\",requestsMem\":\512Mi\",requestsCpu\":\500m}]}");
listSpecValues.put("custom", "{\tenant_id\":\0504201b6c80256b2f08c0099f0c8fe4\",cluster_id
\":\1b2ec02d-a3b2-11ec-b0d0-0255ac100099\",stub_domains\":{\},\upstream_nameservers\":[]}");
InstanceRequestSpec specbody = new InstanceRequestSpec();
specbody.withVersion("1.17.15")
    .withClusterID("1b2ec02d-a3b2-11ec-b0d0-0255ac100099")
    .withValues(listSpecValues)
    .withAddonTemplateName("coredns");
Map<String, String> listMetadataAnnotations = new HashMap<>();
listMetadataAnnotations.put("addon.install/type", "install");
AddonMetadata metadatabody = new AddonMetadata();
metadatabody.withAnnotations(listMetadataAnnotations);
body.withSpec(specbody);
body.withMetadata(metadatabody);
body.withApiVersion("v3");
body.withKind("Addon");
request.withBody(body);
try {
    CreateAddonInstanceResponse response = client.createAddonInstance(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

Install the coredns add-on of version 1.17.15. The add-on specification is 2500 QPS, and the number of add-on instances is 2.

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *
```

```
if __name__ == "__main__":
```

```
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
```

```
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
    ak = __import__('os').getenv("CLOUD_SDK_AK")
```

```
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = BasicCredentials(ak, sk) \

client = CceClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(CceRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = CreateAddonInstanceRequest()
    listValuesSpec = {
        "basic": {"rbac_enabled": true, "swr_user": "hwofficial", "image_version": "1.17.15", "cluster_ip": "10.247.3.10", "platform": "linux-amd64", "swr_addr": "<Replace_SWR_address>"},
        "flavor": {"replicas": 2, "name": "2500", "resources": [{"limitsCpu": "500m", "name": "coredns", "limitsMem": "512Mi", "requestsMem": "512Mi", "requestsCpu": "500m"}]},
        "custom": {"tenant_id": "0504201b6c80256b2f08c0099f0c8fe4", "cluster_id": "1b2ec02d-a3b2-11ec-b0d0-0255ac100099", "stub_domains": {}, "upstream_nameservers": []}
    }
    specbody = InstanceRequestSpec(
        version="1.17.15",
        cluster_id="1b2ec02d-a3b2-11ec-b0d0-0255ac100099",
        values=listValuesSpec,
        addon_template_name="coredns"
    )
    listAnnotationsMetadata = {
        "addon.install/type": "install"
    }
    metadatabody = AddonMetadata(
        annotations=listAnnotationsMetadata
    )
    request.body = InstanceRequest(
        spec=specbody,
        metadata=metadatabody,
        api_version="v3",
        kind="Addon"
    )
    response = client.create_addon_instance(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

Install the coredns add-on of version 1.17.15. The add-on specification is 2500 QPS, and the number of add-on instances is 2.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
```

```

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    Build()

client := cce.NewCceClient(
    cce.CceClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateAddonInstanceRequest{}
var listValuesSpec = map[string]interface{}{
    "basic": "{\\"rbac_enabled\\":true,\\"swr_user\\":\\"hwofficial\\",\\"image_version\\":\\"1.17.15\\",\\"cluster_ip
\\":\\"10.247.3.10\\",\\"platform\\":\\"linux-amd64\\",\\"swr_addr\\":\\"<Replace_SWR_address>\\",
    "flavor": "{\\"replicas\\":2,\\"name\\":2500,\\"resources\\":[{\\"limitsCpu\\":\\"500m\\",\\"name\\":\\"coredns
\\",\\"limitsMem\\":\\"512Mi\\",\\"requestsMem\\":\\"512Mi\\",\\"requestsCpu\\":\\"500m\\"}]},
    "custom": "{\\"tenant_id\\":\\"0504201b6c80256b2f08c0099f0c8fe4\\",\\"cluster_id\\":\\"1b2ec02d-
a3b2-11ec-b0d0-0255ac100099\\",\\"stub_domains\\":{\\"},\\"upstream_nameservers\\":[{\\"}]",
}
versionSpec:= "1.17.15"
specbody := &model.InstanceRequestSpec{
    Version: &versionSpec,
    ClusterID: "1b2ec02d-a3b2-11ec-b0d0-0255ac100099",
    Values: listValuesSpec,
    AddonTemplateName: "coredns",
}
var listAnnotationsMetadata = map[string]string{
    "addon.install/type": "install",
}
metadatabody := &model.AddonMetadata{
    Annotations: listAnnotationsMetadata,
}
request.Body = &model.InstanceRequest{
    Spec: specbody,
    Metadata: metadatabody,
    ApiVersion: "v3",
    Kind: "Addon",
}
response, err := client.CreateAddonInstance(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
201	OK

Error Codes

See [Error Codes](#).

4.6.2 Listing Add-on Templates

Function

This API is used to query add-on templates.

Calling Method

For details, see [Calling APIs](#).

URI

GET /api/v3/addontemplates

Table 4-710 Query Parameters

Parameter	Mandatory	Type	Description
addon_template_name	No	String	Add-on name or alias for search. If this parameter is not specified, all the add-ons will be searched.

Request Parameters

Table 4-711 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-712 Response body parameters

Parameter	Type	Description
kind	String	API type. The value is fixed at Addon and cannot be changed.
apiVersion	String	API version. The value is fixed at v3 and cannot be changed.
items	Array of AddonTemplate objects	List of add-on templates.

Table 4-713 AddonTemplate

Parameter	Type	Description
kind	String	API type. The value is fixed at Addon and cannot be changed.
apiVersion	String	API version. The value is fixed at v3 and cannot be changed.
metadata	AddonMetadata object	Basic information about the object. Metadata is a collection of attributes.
spec	TemplateSpec object	Detailed description of the add-on template.

Table 4-714 AddonMetadata

Parameter	Type	Description
uid	String	Unique ID
name	String	Add-on name
alias	String	Add-on alias
labels	Map<String,String>	Add-on labels in key-value pairs. This is a reserved field and does not take effect.
annotations	Map<String,String>	Add-on annotations in the format of key-value pairs. <ul style="list-style-type: none"> For add-on installation, the value is fixed at {"addon.install/type":"install"}. For add-on upgrade, the value is fixed at {"addon.upgrade/type":"upgrade"}.
updateTimestamp	String	Updated at

Parameter	Type	Description
creationTimes tamp	String	Created at

Table 4-715 Templatespec

Parameter	Type	Description
type	String	Template type. The value is helm or static .
require	Boolean	Whether the add-on is mandatory.
labels	Array of strings	Group to which the template belongs.
logoURL	String	URL of the logo image.
readmeURL	String	Add-on details and usage description.
description	String	Description
versions	Array of Versions objects	Template version details.

Table 4-716 Versions

Parameter	Type	Description
version	String	Add-on version.
input	Object	Add-on installation parameters.
stable	Boolean	Whether the add-on version is a stable release.
translate	Object	Translation information used by the GUI.
supportVersions	Array of SupportVersions objects	Cluster versions that support the add-on.
creationTimes tamp	String	Creation time.
updateTime stamp	String	Update time.

Table 4-717 SupportVersions

Parameter	Type	Description
clusterType	String	Cluster type that supports the add-on.
clusterVersion	Array of strings	Cluster versions that support the add-on. The value is a regular expression.

Example Requests

None

Example Responses

Status code: 200

OK

```
{
  "kind": "Addon",
  "apiVersion": "v3",
  "items": [ {
    "kind": "Addon",
    "apiVersion": "v3",
    "metadata": {
      "uid": "coredns",
      "name": "coredns",
      "alias": "coredns",
      "creationTimestamp": "2018-11-04T16:15:56Z",
      "updateTimestamp": "2022-01-11T14:32:10Z"
    },
    "spec": {
      "type": "helm",
      "require": true,
      "labels": [ "ServiceDiscovery" ],
      "logoURL": "",
      "description": "CoreDNS is a DNS server that chains plugins and provides Kubernetes DNS Services",
      "versions": [ {
        "version": "1.13.6",
        "input": {
          "basic": {
            "cluster_ip": "10.247.3.10",
            "ipv6": false,
            "platform": "linux-amd64",
            "swr_addr": "100.125.16.65:20202",
            "swr_user": "hwofficial"
          },
          "parameters": {
            "custom": {
              "stub_domains": "",
              "upstream_nameservers": ""
            },
            "flavor1": {
              "name": 2500,
              "replicas": 2,
              "resources": [ {
                "limitsCpu": "500m",
                "limitsMem": "512Mi",
                "name": "coredns",
                "requestsCpu": "500m",
                "requestsMem": "512Mi"
              } ]
            }
          }
        }
      } ]
    }
  } ]
}
```



```

"flavor2" : {
  "name" : 5000,
  "replicas" : 2,
  "resources" : [ {
    "limitsCpu" : "1000m",
    "limitsMem" : "1024Mi",
    "name" : "coredns",
    "requestsCpu" : "1000m",
    "requestsMem" : "1024Mi"
  } ]
},
"flavor3" : {
  "name" : 10000,
  "replicas" : 2,
  "resources" : [ {
    "limitsCpu" : "2000m",
    "limitsMem" : "2048Mi",
    "name" : "coredns",
    "requestsCpu" : "2000m",
    "requestsMem" : "2048Mi"
  } ]
},
"flavor4" : {
  "name" : 20000,
  "replicas" : 4,
  "resources" : [ {
    "limitsCpu" : "2000m",
    "limitsMem" : "2048Mi",
    "name" : "coredns",
    "requestsCpu" : "2000m",
    "requestsMem" : "2048Mi"
  } ]
}
},
"stable" : true,
"translate" : {
  "en_US" : {
    "addon" : {
      "changeLog" : "Support for clusters with new version",
      "description" : "CoreDNS is a DNS server that chains plugins and provides Kubernetes DNS
Services"
    },
    "description" : {
      "Parameters.custom.stub_domains" : "The target nameserver may itself be a Kubernetes service.
For instance, you can run your own copy of dnsmasq to export custom DNS names into the ClusterDNS
namespace, a JSON map using a DNS suffix key (e.g. \"acme.local\") and a value consisting of a JSON array
of DNS IPs.",
      "Parameters.custom.upstream_nameservers" : "If specified, then the values specified replace the
nameservers taken by default from the node's /etc/resolv.conf. Limits:a maximum of three upstream
nameservers can be specified, A JSON array of DNS IPs.",
      "Parameters.flavor1.description" : "Concurrent domain name resolution ability - External domain
name: 2500 qps, Internal domain name: 10000 qps",
      "Parameters.flavor1.name" : 2500,
      "Parameters.flavor2.description" : "Concurrent domain name resolution ability - External domain
name: 5000 qps, Internal domain name: 20000 qps",
      "Parameters.flavor2.name" : 5000,
      "Parameters.flavor3.description" : "Concurrent domain name resolution ability - External domain
name: 10000 qps, Internal domain name: 40000 qps",
      "Parameters.flavor3.name" : 10000,
      "Parameters.flavor4.description" : "Concurrent domain name resolution ability - External domain
name: 20000 qps, Internal domain name: 80000 qps",
      "Parameters.flavor4.name" : 20000
    },
    "key" : {
      "Parameters.custom.stub_domains" : "stub domain",
      "Parameters.custom.upstream_nameservers" : "upstream nameservers"
    }
  }
},

```

```

"fr_FR" : {
  "addon" : {
    "changeLog" : "Prise en charge des clusters avec une nouvelle version",
    "description" : "Un serveur DNS qui enchaîne les plug-ins et fournit des services DNS Kubernetes."
  },
  "description" : {
    "Parameters.custom.stub_domains" : "Le serveur de noms cible peut lui-même être un service
Kubernetes. Par exemple, vous pouvez exécuter votre propre copie de dnsmasq pour exporter des noms
DNS personnalisés dans l'espace de noms ClusterDNS, une carte JSON à l'aide d'une clé de suffixe DNS (par
exemple, «acme.local») et une valeur constituée d'un tableau JSON d'adresses IP DNS.",
    "Parameters.custom.upstream_nameservers" : "Si spécifié, les valeurs spécifiées remplacent les
serveurs de noms pris par défaut dans le fichier /etc/resolv.conf du nœud. Limites: un maximum de trois
serveurs de noms en amont peuvent être spécifiés, un tableau JSON d'adresses IP DNS.",
    "Parameters.flavor1.description" : "Capacité de résolution de nom de domaine simultanée - Nom
de domaine externe: 2500 qps, Nom de domaine interne: 10000 qp",
    "Parameters.flavor1.name" : 2500,
    "Parameters.flavor2.description" : "Capacité de résolution de nom de domaine simultanée - Nom
de domaine externe: 5000 qps, Nom de domaine interne: 20000 qp",
    "Parameters.flavor2.name" : 5000,
    "Parameters.flavor3.description" : "Capacité de résolution de nom de domaine simultanée - Nom
de domaine externe: 10000 qps, Nom de domaine interne: 40000 qp",
    "Parameters.flavor3.name" : 10000,
    "Parameters.flavor4.description" : "Capacité de résolution de nom de domaine simultanée - Nom
de domaine externe: 20000 qps, Nom de domaine interne: 80000 qp",
    "Parameters.flavor4.name" : 20000
  },
  "key" : {
    "Parameters.custom.stub_domains" : "domaine stub",
    "Parameters.custom.upstream_nameservers" : "serveurs de noms en amont"
  }
},
"zh_CN" : {
  "addon" : {
    "changeLog" : "",
    "description" : ""
  },
  "description" : {
    "Parameters.custom.stub_domains" : "",
    "Parameters.custom.upstream_nameservers" : "",
    "Parameters.flavor1.description" : "",
    "Parameters.flavor1.name" : 2500,
    "Parameters.flavor2.description" : "",
    "Parameters.flavor2.name" : 5000,
    "Parameters.flavor3.description" : "",
    "Parameters.flavor3.name" : 10000,
    "Parameters.flavor4.description" : "",
    "Parameters.flavor4.name" : 20000
  },
  "key" : {
    "Parameters.custom.stub_domains" : "",
    "Parameters.custom.upstream_nameservers" : ""
  }
}
},
"supportVersions" : [ {
  "clusterType" : "VirtualMachine",
  "clusterVersion" : [ "v1.13.*" ]
}, {
  "clusterType" : "BareMetal",
  "clusterVersion" : [ "v1.13.*" ]
}, {
  "clusterType" : "ARM64",
  "clusterVersion" : [ "v1.13.*" ]
} ],
"creationTimestamp" : "2021-03-18T12:51:05Z",
"updateTimestamp" : "2021-03-18T12:51:05Z"
} ]
}

```

```
    }  
  }  
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cce.v3.region.CceRegion;  
import com.huaweicloud.sdk.cce.v3.*;  
import com.huaweicloud.sdk.cce.v3.model.*;  
  
public class ListAddonTemplatesSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
  
        ICredential auth = new BasicCredentials()  
            .withAk(ak)  
            .withSk(sk);  
  
        CceClient client = CceClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListAddonTemplatesRequest request = new ListAddonTemplatesRequest();  
        try {  
            ListAddonTemplatesResponse response = client.listAddonTemplates(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

Python

```
# coding: utf-8  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkcce.v3.region.cce_region import CceRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkcce.v3 import *
```

```

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListAddonTemplatesRequest()
        response = client.list_addon_templates(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)

```

Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAddonTemplatesRequest{}
    response, err := client.ListAddonTemplates(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	OK

Error Codes

See [Error Codes](#).

4.6.3 Updating an Add-on Instance

Function

This API is used to update an add-on instance.

Calling Method

For details, see [Calling APIs](#).

URI

PUT /api/v3/addons/{id}

Table 4-718 Path Parameters

Parameter	Mandatory	Type	Description
id	Yes	String	Add-on instance ID.

Request Parameters

Table 4-719 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Table 4-720 Request body parameters

Parameter	Mandatory	Type	Description
kind	Yes	String	API type. The value is fixed at Addon and cannot be changed. Any user-defined value is invalid.
apiVersion	Yes	String	API version. The value is fixed at v3 and cannot be changed. Any user-defined value is invalid.
metadata	Yes	AddonMetadata object	Basic information about the object. Metadata is a collection of attributes.
spec	Yes	InstanceRequestSpec object	Detailed description of add-on installation or upgrade.

Table 4-721 AddonMetadata

Parameter	Mandatory	Type	Description
uid	No	String	Unique ID
name	No	String	Add-on name
alias	No	String	Add-on alias
labels	No	Map<String,String>	Add-on labels in key-value pairs. This is a reserved field and does not take effect.

Parameter	Mandatory	Type	Description
annotations	No	Map<String,String>	Add-on annotations in the format of key-value pairs. <ul style="list-style-type: none"> For add-on installation, the value is fixed at {"addon.install/type":"install"}. For add-on upgrade, the value is fixed at {"addon.upgrade/type":"upgrade"}.
updateTimestamp	No	String	Updated at
creationTimestamp	No	String	Created at

Table 4-722 InstanceRequestSpec

Parameter	Mandatory	Type	Description
version	No	String	Version of the add-on to install or upgrade, for example, 1.0.0 . <ul style="list-style-type: none"> Installation: This parameter is optional. If not specified, the latest version supported by the cluster is used. Upgrade: This parameter is mandatory. The version number must be specified.
clusterID	Yes	String	Cluster ID.
values	Yes	Map<String,Object>	Add-on template installation parameters (varying depending on the add-on). During the add-on upgrade, you need to specify all the installation parameters. If the parameters are not specified, the default values in the add-on template are used. The current add-on installation parameters can be obtained through the API for querying add-on instances.

Parameter	Mandatory	Type	Description
addonTemplate	Yes	String	Name of the add-on template to be installed, for example, coredns .

Response Parameters

Status code: 200

Table 4-723 Response body parameters

Parameter	Type	Description
kind	String	API type. The value is fixed at Addon and cannot be changed.
apiVersion	String	API version. The value is fixed at v3 and cannot be changed.
metadata	AddonMetadata object	Basic information about the object. Metadata is a collection of attributes.
spec	InstanceSpec object	Detailed description of the add-on instance.
status	AddonInstanceStatus object	Add-on instance status.

Table 4-724 AddonMetadata

Parameter	Type	Description
uid	String	Unique ID
name	String	Add-on name
alias	String	Add-on alias
labels	Map<String,String>	Add-on labels in key-value pairs. This is a reserved field and does not take effect.
annotations	Map<String,String>	Add-on annotations in the format of key-value pairs. <ul style="list-style-type: none"> For add-on installation, the value is fixed at {"addon.install/type":"install"}. For add-on upgrade, the value is fixed at {"addon.upgrade/type":"upgrade"}.

Parameter	Type	Description
updateTimestamp	String	Updated at
creationTimestamp	String	Created at

Table 4-725 InstanceSpec

Parameter	Type	Description
clusterID	String	Cluster ID.
version	String	Add-on template version, for example, 1.0.0 .
addonTemplateName	String	Add-on template name, for example, coredns .
addonTemplateType	String	Add-on template type.
addonTemplateLogo	String	URL for obtaining the add-on template logo.
addonTemplateLabels	Array of strings	Labels of the add-on template.
description	String	Add-on template description.
values	Map<String,Object>	Add-on template installation parameters (varying depending on the add-on). Set the parameters accordingly.

Table 4-726 AddonInstanceStatus

Parameter	Type	Description
status	String	<p>Add-on instance status. Options:</p> <ul style="list-style-type: none"> ● running: All of the add-on instances are running. This specifies that the add-on runs properly. ● abnormal: The add-on instances are abnormal and the add-on cannot be used. You can click the add-on name to view exceptions. ● installing: The add-on is being installed. ● installFailed: Installing the add-on failed. In this case, uninstall the add-on and then reinstall it. ● upgrading: The add-on is being upgraded. ● upgradeFailed: Upgrading the add-on failed. In this case, upgrade the add-on again or uninstall the add-on and reinstall it. ● deleting: The add-on is being deleted. ● deleteFailed: Deleting the add-on failed. In this case, uninstall the add-on again. ● deleteSuccess: Deleting the add-on succeeded. ● available: Only some instances of the add-on are running. This specifies that some functions of the add-on are available. ● rollbacking: The add-on is being rolled back. ● rollbackFailed: Rolling back the add-on failed. In this case, roll back the add-on again or uninstall the add-on and reinstall it. ● unknown: The add-on chart instance does not exist.
Reason	String	Cause of the add-on installation failure.
message	String	Installation error details.
targetVersions	Array of strings	Versions to which the current add-on version can be upgraded.
currentVersion	Versions object	Information about the current add-on version.
isRollbackable	Boolean	Whether the add-on version can be rolled back to the source version.

Parameter	Type	Description
previousVersion	String	The add-on version before upgrade or rollback

Table 4-727 Versions

Parameter	Type	Description
version	String	Add-on version.
input	Object	Add-on installation parameters.
stable	Boolean	Whether the add-on version is a stable release.
translate	Object	Translation information used by the GUI.
supportVersions	Array of SupportVersions objects	Cluster versions that support the add-on.
creationTimestamp	String	Creation time.
updateTimestamp	String	Update time.

Table 4-728 SupportVersions

Parameter	Type	Description
clusterType	String	Cluster type that supports the add-on.
clusterVersion	Array of strings	Cluster versions that support the add-on. The value is a regular expression.

Example Requests

Update the everest add-on to version 2.1.30.

```
{
  "kind": "Addon",
  "apiVersion": "v3",
  "metadata": {
    "annotations": {
      "addon.upgrade/type": "upgrade"
    }
  },
  "spec": {
    "clusterID": "*****",
    "version": "2.1.30",
    "addonTemplateName": "everest",
    "values": {
      "basic": {
```

```
"bms_url" : "*****",
"driver_init_image_version" : "2.1.30",
"ecsEndpoint" : "*****",
"everest_image_version" : "2.1.30",
"evs_url" : "*****",
"iam_url" : "*****",
"ims_url" : "*****",
"obs_url" : "*****",
"platform" : "linux-amd64",
"sfs30_url" : "*****",
"sfs_turbo_url" : "*****",
"sfs_url" : "*****",
"supportHcs" : false,
"swr_addr" : "*****",
"swr_user" : "hwofficial",
"rbac_enabled" : true,
"cluster_version" : "v1.23"
},
"flavor" : {
  "description" : "High available",
  "name" : "HA",
  "replicas" : 2,
  "resources" : [ {
    "limitsCpu" : "250m",
    "limitsMem" : "2000Mi",
    "name" : "everest-csi-controller",
    "requestsCpu" : "250m",
    "requestsMem" : "1500Mi"
  }, {
    "limitsCpu" : "500m",
    "limitsMem" : "300Mi",
    "name" : "everest-csi-driver",
    "requestsCpu" : "100m",
    "requestsMem" : "300Mi"
  } ],
  "category" : [ "CCE", "Turbo" ]
},
"custom" : {
  "cluster_id" : "*****",
  "csi_attacher_detach_worker_threads" : "60",
  "csi_attacher_worker_threads" : "60",
  "default_vpc_id" : "*****",
  "disable_auto_mount_secret" : false,
  "enable_node_attacher" : true,
  "flow_control" : { },
  "multiAZEnabled" : false,
  "over_subscription" : "80",
  "project_id" : "*****",
  "volume_attaching_flow_ctrl" : "0"
}
}
}
```

Example Responses

Status code: 200

OK

```
{
  "kind" : "Addon",
  "apiVersion" : "v3",
  "metadata" : {
    "uid" : "*****",
    "name" : "everest",
    "alias" : "everest",
    "creationTimestamp" : "2023-07-03T10:57:43Z",
    "updateTimestamp" : "2023-07-03T11:24:05Z"
  }
}
```

```

},
"spec" : {
  "clusterID" : "*****",
  "version" : "2.1.30",
  "addonTemplateName" : "everest",
  "addonTemplateType" : "helm",
  "addonTemplateLogo" : "*****",
  "addonTemplateLabels" : [ "Storage" ],
  "description" : "",
  "values" : {
    "basic" : {
      "bms_url" : "*****",
      "cluster_version" : "v1.23",
      "driver_init_image_version" : "2.1.30",
      "ecsEndpoint" : "*****",
      "everest_image_version" : "2.1.30",
      "evs_url" : "*****",
      "iam_url" : "*****",
      "ims_url" : "*****",
      "obs_url" : "*****",
      "platform" : "linux-amd64",
      "rbac_enabled" : true,
      "sfs30_url" : "*****",
      "sfs_turbo_url" : "*****",
      "sfs_url" : "*****",
      "supportHcs" : false,
      "swr_addr" : "*****",
      "swr_user" : "hwofficial"
    },
    "custom" : {
      "cluster_id" : "*****",
      "cluster_version" : "v1.23.8-r0",
      "csi_attacher_detach_worker_threads" : "60",
      "csi_attacher_worker_threads" : "60",
      "default_vpc_id" : "*****",
      "disable_auto_mount_secret" : false,
      "enable_node_attacher" : false,
      "flow_control" : { },
      "multiAZEnabled" : false,
      "over_subscription" : "80",
      "project_id" : "*****",
      "volume_attaching_flow_ctrl" : "0"
    },
    "flavor" : {
      "category" : [ "CCE", "Turbo" ],
      "description" : "Has only one instance",
      "name" : "Single",
      "replicas" : 1,
      "resources" : [ {
        "limitsCpu" : "250m",
        "limitsMem" : "600Mi",
        "name" : "everest-csi-controller",
        "requestsCpu" : "250m",
        "requestsMem" : "600Mi"
      }, {
        "limitsCpu" : "100m",
        "limitsMem" : "300Mi",
        "name" : "everest-csi-driver",
        "requestsCpu" : "100m",
        "requestsMem" : "300Mi"
      } ]
    },
    "systemAutoInject" : {
      "cluster" : {
        "clusterID" : "*****",
        "clusterNetworkMode" : "vpc-router",
        "clusterVersion" : "v1.23.8-r0"
      },
      "user" : {

```

```

    "projectID" : "*****"
  }
}
},
"status" : {
  "status" : "upgrading",
  "Reason" : "addon upgrading",
  "message" : "",
  "targetVersions" : null,
  "isRollbackable" : false,
  "currentVersion" : {
    "version" : "2.1.30",
    "input" : {
      "basic" : {
        "bms_url" : "*****",
        "driver_init_image_version" : "2.1.30",
        "ecsEndpoint" : "*****",
        "everest_image_version" : "2.1.30",
        "evs_url" : "*****",
        "iam_url" : "*****",
        "ims_url" : "*****",
        "obs_url" : "*****",
        "platform" : "*****",
        "sfs30_url" : "*****",
        "sfs_turbo_url" : "*****",
        "sfs_url" : "*****",
        "supportHcs" : false,
        "swr_addr" : "*****",
        "swr_user" : "hwofficial"
      },
      "parameters" : {
        "common" : {
          "defaultVPCId" : 1234567
        },
        "custom" : {
          "cluster_id" : "",
          "csi_attacher_detach_worker_threads" : "60",
          "csi_attacher_worker_threads" : "60",
          "default_vpc_id" : "",
          "disable_auto_mount_secret" : false,
          "enable_node_attacher" : false,
          "flow_control" : { },
          "multiAZEnabled" : false,
          "over_subscription" : "80",
          "project_id" : "",
          "volume_attaching_flow_ctrl" : "0"
        },
        "flavor1" : {
          "description" : "High available",
          "name" : "HA",
          "replicas" : 2,
          "resources" : [ {
            "limitsCpu" : "250m",
            "limitsMem" : "1500Mi",
            "name" : "everest-csi-controller",
            "requestsCpu" : "250m",
            "requestsMem" : "600Mi"
          }, {
            "limitsCpu" : "500m",
            "limitsMem" : "300Mi",
            "name" : "everest-csi-driver",
            "requestsCpu" : "100m",
            "requestsMem" : "300Mi"
          } ]
        },
        "flavor2" : {
          "description" : "Has only one instance",
          "name" : "Single",

```

```

    "replicas" : 1,
    "resources" : [ {
      "limitsCpu" : "250m",
      "limitsMem" : "600Mi",
      "name" : "everest-csi-controller",
      "requestsCpu" : "250m",
      "requestsMem" : "600Mi"
    }, {
      "limitsCpu" : "100m",
      "limitsMem" : "300Mi",
      "name" : "everest-csi-driver",
      "requestsCpu" : "100m",
      "requestsMem" : "300Mi"
    } ]
  },
  "flavor3" : {
    "description" : "custom resources",
    "name" : "custom-resources",
    "replicas" : 2,
    "resources" : [ {
      "limitsCpu" : "250m",
      "limitsMem" : "2000Mi",
      "name" : "everest-csi-controller",
      "requestsCpu" : "250m",
      "requestsMem" : "1500Mi"
    }, {
      "limitsCpu" : "500m",
      "limitsMem" : "300Mi",
      "name" : "everest-csi-driver",
      "requestsCpu" : "100m",
      "requestsMem" : "300Mi"
    } ]
  }
},
"stable" : true,
"translate" : {
  "en_US" : {
    "addon" : {
      "changeLog" : "",
      "description" : ""
    },
    "description" : {
      "Parameters.flavor1.description" : "Deploy the add-on with two instances, delivering high
availability but requiring more compute resources.",
      "Parameters.flavor1.name" : "HA",
      "Parameters.flavor2.description" : "Deploy the add-on with one instance.",
      "Parameters.flavor2.name" : "Standalone",
      "Parameters.flavor3.name" : "Custom"
    }
  },
  "fr_FR" : {
    "addon" : {
      "changeLog" : "",
      "description" : ""
    },
    "description" : {
      "Parameters.flavor1.description" : "Déployez avec deux instances, haute disponibilité.",
      "Parameters.flavor1.name" : "HA",
      "Parameters.flavor2.description" : "Déployez avec une seule instance.",
      "Parameters.flavor2.name" : "Célibataire",
      "Parameters.flavor3.name" : "Douane"
    }
  },
  "zh_CN" : {
    "addon" : {
      "changeLog" : "",
      "description" : ""
    },
  },

```



```
\,\"everest_image_version\": \"2.1.30\", \"swr_addr\": \"*****\"});
    listSpecValues.put(\"flavor\", \"{\\\"replicas\\\":2,\\\"name\\\":\\\"HA\\\",\\\"description\\\":\\\"High available
\\,\"resources\\\":[{\\\"limitsCpu\\\":\\\"250m\\\",\\\"name\\\":\\\"everest-csi-controller\\\",\\\"limitsMem\\\":\\\"2000Mi
\\,\"requestsMem\\\":\\\"1500Mi\\\",\\\"requestsCpu\\\":\\\"250m\\\"},{\\\"limitsCpu\\\":\\\"500m\\\",\\\"name\\\":\\\"everest-csi-
driver\\\",\\\"limitsMem\\\":\\\"300Mi\\\",\\\"requestsMem\\\":\\\"300Mi\\\",\\\"requestsCpu\\\":\\\"100m\\\"}],\\\"category\\\":[\\\"CCE
\\,\"Turbo\\\"]}");
    listSpecValues.put(\"custom\", \"{\\\"csi_attacher_worker_threads\\\":\\\"60\\\",\\\"cluster_id\\\":\\\"*****
\\,\"csi_attacher_detach_worker_threads\\\":\\\"60\\\",\\\"disable_auto_mount_secret\\\":false,\\\"over_subscription
\\\":\\\"80\\\",\\\"project_id\\\":\\\"*****\\\",\\\"enable_node_attacher\\\":true,\\\"volume_attaching_flow_ctrl
\\\":\\\"0\\\",\\\"multiAZEnabled\\\":false,\\\"flow_control\\\":{\\\"default_vpc_id\\\":\\\"*****\\\"}}");
    InstanceRequestSpec specbody = new InstanceRequestSpec();
    specbody.withVersion(\"2.1.30")
        .withClusterID(\"*****")
        .withValues(listSpecValues)
        .withAddonTemplateName(\"everest");
    Map<String, String> listMetadataAnnotations = new HashMap<>();
    listMetadataAnnotations.put(\"addon.upgrade/type\", \"upgrade");
    AddonMetadata metadatabody = new AddonMetadata();
    metadatabody.withAnnotations(listMetadataAnnotations);
    body.withSpec(specbody);
    body.withMetadata(metadatabody);
    body.withApiVersion(\"v3");
    body.withKind(\"Addon");
    request.withBody(body);
    try {
        UpdateAddonInstanceResponse response = client.updateAddonInstance(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

Update the everest add-on to version 2.1.30.

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *
```

```
if __name__ == \"__main__\":
```

```
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
```

```
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
    ak = __import__('os').getenv(\"CLOUD_SDK_AK\")
    sk = __import__('os').getenv(\"CLOUD_SDK_SK\")
```

```
    credentials = BasicCredentials(ak, sk) \
```

```
    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of(\"<YOUR REGION>\")) \
        .build()
```

```
    try:
```

```

request = UpdateAddonInstanceRequest()
request.id = "{id}"
listValuesSpec = {
    "basic": "{\
"evs_url\":"*****",\
"obs_url\":"*****",\
"cluster_version\":"v1.23",\
"supportHcs":false,\
"iam_url\":"*****",\
"rbac_enabled\":true,\
"ecsEndpoint\":"*****",\
"sfs_url\":"*****",\
"platform\":"linux-amd64",\
"ims_url\":"*****",\
"driver_init_image_version\":"2.1.30",\
"sfs30_url\":"*****",\
"sfs_turbo_url\":"*****",\
"swr_user\":"hwofficial",\
"bms_url\":"*****",\
"everest_image_version\":"2.1.30",\
"swr_addr\":"*****"}",
    "flavor": "{\
"replicas\":2,\
"name\":"HA",\
"description\":"High available",\
"resources":[\
{"limitsCpu\":"250m",\
"name\":"everest-csi-controller",\
"limitsMem\":"2000Mi",\
"requestsMem\":"1500Mi",\
"requestsCpu\":"250m"},\
{"limitsCpu\":"500m",\
"name\":"everest-csi-driver",\
"limitsMem\":"300Mi",\
"requestsMem\":"300Mi",\
"requestsCpu\":"100m"}],\
"category\":["CCE",\
"Turbo"]},\
    "custom": "{\
"csi_attacher_worker_threads\":"60",\
"cluster_id\":"*****",\
"csi_attacher_detach_worker_threads\":"60",\
"disable_auto_mount_secret\":false,\
"over_subscription\":"80",\
"project_id\":"*****",\
"enable_node_attacher\":true,\
"volume_attaching_flow_ctrl\":"0",\
"multiAZEnabled\":false,\
"flow_control\":{\},\
"default_vpc_id\":"*****"}"
}
specbody = InstanceRequestSpec(
    version="2.1.30",
    cluster_id="*****",
    values=listValuesSpec,
    addon_template_name="everest"
)
listAnnotationsMetadata = {
    "addon.upgrade/type": "upgrade"
}
metadatabody = AddonMetadata(
    annotations=listAnnotationsMetadata
)
request.body = InstanceRequest(
    spec=specbody,
    metadata=metadatabody,
    api_version="v3",
    kind="Addon"
)
response = client.update_addon_instance(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)

```

Go

Update the everest add-on to version 2.1.30.

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).

```

```

Build()

client := cce.NewCceClient(
    cce.CceClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateAddonInstanceRequest{}
request.Id = "{id}"
var listValuesSpec = map[string]interface{}{
    "basic": "{\
\"evs_url\": \"*****\",
\"obs_url\": \"*****\",
\"cluster_version\": \"v1.23\",
\"supportHcs\": false,
\"iam_url\": \"*****\",
\"rbac_enabled\": true,
\"ecsEndpoint\": \"*****\",
\"sfs_url\": \"*****\",
\"platform\": \"linux-amd64\",
\"ims_url\": \"*****\",
\"driver_init_image_version\": \"2.1.30\",
\"sfs30_url\": \"*****\",
\"sfs_turbo_url\": \"*****\",
\"swr_user\": \"hwofficial\",
\"bms_url\": \"*****\",
\"everest_image_version\": \"2.1.30\",
\"swr_addr\": \"*****\"}",
    "flavor": "{\
\"replicas\": 2,
\"name\": \"HA\",
\"description\": \"High available\",
\"resources\": {\
\"limitsCpu\": \"250m\",
\"name\": \"everest-csi-controller\",
\"limitsMem\": \"2000Mi\",
\"requestsMem\": \"1500Mi\",
\"requestsCpu\": \"250m\"},
{\
\"limitsCpu\": \"500m\",
\"name\": \"everest-csi-driver\",
\"limitsMem\": \"300Mi\",
\"requestsMem\": \"300Mi\",
\"requestsCpu\": \"100m\"}},
\"category\": [\"CCE\", \"Turbo\"]",
    "custom": "{\
\"csi_attacher_worker_threads\": \"60\",
\"cluster_id\": \"*****\",
\"csi_attacher_detach_worker_threads\": \"60\",
\"disable_auto_mount_secret\": false,
\"over_subscription\": \"80\",
\"project_id\": \"*****\",
\"enable_node_attacher\": true,
\"volume_attaching_flow_ctrl\": \"0\",
\"multiAZEnabled\": false,
\"flow_control\": {},
\"default_vpc_id\": \"*****\"}",
}
versionSpec := "2.1.30"
specbody := &model.InstanceRequestSpec{
    Version: &versionSpec,
    ClusterID: "*****",
    Values: listValuesSpec,
    AddonTemplateName: "everest",
}
var listAnnotationsMetadata = map[string]string{
    "addon.upgrade/type": "upgrade",
}
metadatabody := &model.AddonMetadata{
    Annotations: listAnnotationsMetadata,
}
request.Body = &model.InstanceRequest{
    Spec: specbody,
    Metadata: metadatabody,
    ApiVersion: "v3",
    Kind: "Addon",
}
response, err := client.UpdateAddonInstance(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	OK

Error Codes

See [Error Codes](#).

4.6.4 Rolling Back an Add-on Instance

Function

Roll back the add-on instance version to the source version. Only when the **status.isRollbackable** parameter is set to **true**, and the add-on instance is in the **running**, **available**, **abnormal**, **upgradeFailed**, or **rollbackFailed** state, the version can be rolled back to the source version.

Calling Method

For details, see [Calling APIs](#).

URI

POST /api/v3/addons/{id}/operation/rollback

Table 4-729 Path Parameters

Parameter	Mandatory	Type	Description
id	Yes	String	Add-on instance ID

Request Parameters

Table 4-730 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Table 4-731 Request body parameters

Parameter	Mandatory	Type	Description
clusterID	Yes	String	Cluster ID

Response Parameters

Status code: 200

Table 4-732 Response body parameters

Parameter	Type	Description
kind	String	API type. The value is fixed at Addon and cannot be changed.
apiVersion	String	API version. The value is fixed at v3 and cannot be changed.
metadata	AddonMetadata object	Basic information about the object. Metadata is a collection of attributes.
spec	InstanceSpec object	Detailed description of the add-on instance.
status	AddonInstanceStatus object	Add-on instance status.

Table 4-733 AddonMetadata

Parameter	Type	Description
uid	String	Unique ID
name	String	Add-on name
alias	String	Add-on alias
labels	Map<String,String>	Add-on labels in key-value pairs. This is a reserved field and does not take effect.
annotations	Map<String,String>	Add-on annotations in the format of key-value pairs. <ul style="list-style-type: none"> For add-on installation, the value is fixed at {"addon.install/type":"install"}. For add-on upgrade, the value is fixed at {"addon.upgrade/type":"upgrade"}.
updateTimestamp	String	Updated at
creationTimestamp	String	Created at

Table 4-734 InstanceSpec

Parameter	Type	Description
clusterID	String	Cluster ID.
version	String	Add-on template version, for example, 1.0.0 .
addonTemplate Name	String	Add-on template name, for example, coredns .
addonTemplate Type	String	Add-on template type.
addonTemplate Logo	String	URL for obtaining the add-on template logo.
addonTemplate Labels	Array of strings	Labels of the add-on template.
description	String	Add-on template description.
values	Map<String, Object>	Add-on template installation parameters (varying depending on the add-on). Set the parameters accordingly.

Table 4-735 AddonInstanceStatus

Parameter	Type	Description
status	String	<p>Add-on instance status. Options:</p> <ul style="list-style-type: none"> ● running: All of the add-on instances are running. This specifies that the add-on runs properly. ● abnormal: The add-on instances are abnormal and the add-on cannot be used. You can click the add-on name to view exceptions. ● installing: The add-on is being installed. ● installFailed: Installing the add-on failed. In this case, uninstall the add-on and then reinstall it. ● upgrading: The add-on is being upgraded. ● upgradeFailed: Upgrading the add-on failed. In this case, upgrade the add-on again or uninstall the add-on and reinstall it. ● deleting: The add-on is being deleted. ● deleteFailed: Deleting the add-on failed. In this case, uninstall the add-on again. ● deleteSuccess: Deleting the add-on succeeded. ● available: Only some instances of the add-on are running. This specifies that some functions of the add-on are available. ● rollbacking: The add-on is being rolled back. ● rollbackFailed: Rolling back the add-on failed. In this case, roll back the add-on again or uninstall the add-on and reinstall it. ● unknown: The add-on chart instance does not exist.
Reason	String	Cause of the add-on installation failure.
message	String	Installation error details.
targetVersions	Array of strings	Versions to which the current add-on version can be upgraded.
currentVersion	Versions object	Information about the current add-on version.
isRollbackable	Boolean	Whether the add-on version can be rolled back to the source version.

Parameter	Type	Description
previousVersion	String	The add-on version before upgrade or rollback

Table 4-736 Versions

Parameter	Type	Description
version	String	Add-on version.
input	Object	Add-on installation parameters.
stable	Boolean	Whether the add-on version is a stable release.
translate	Object	Translation information used by the GUI.
supportVersions	Array of SupportVersions objects	Cluster versions that support the add-on.
creationTimestamp	String	Creation time.
updateTimestamp	String	Update time.

Table 4-737 SupportVersions

Parameter	Type	Description
clusterType	String	Cluster type that supports the add-on.
clusterVersion	Array of strings	Cluster versions that support the add-on. The value is a regular expression.

Example Requests

```
{
  "clusterID": "*****"
}
```

Example Responses

Status code: 200

Add-on instance version is rolled back.

```
{
  "kind": "Addon",
  "apiVersion": "v3",
  "metadata": {
    "uid": "*****",
  }
}
```



```

"name" : "everest",
"alias" : "everest",
"creationTimestamp" : "2023-03-15T02:48:01Z",
"updateTimestamp" : "2023-03-15T04:18:45Z"
},
"spec" : {
"clusterID" : "*****",
"version" : "2.1.16",
"addonTemplateName" : "everest",
"addonTemplateType" : "helm",
"addonTemplateLogo" : "*****",
"addonTemplateLabels" : [ "Storage" ],
"description" : "Everest is a cloud native container storage system based on CSI, used\erto support cloud
storages services for Kubernetes",
"values" : {
"basic" : {
"base_image" : "euleros",
"bms_url" : "*****",
"cluster_version" : "v1.25",
"driver_init_image_version" : "2.1.16",
"ecsEndpoint" : "*****",
"euleros_version" : "2.2.5",
"everest_image_version" : "2.1.16",
"evs_url" : "*****",
"iam_url" : "*****",
"ims_url" : "*****",
"obs_url" : "*****",
"platform" : "linux-amd64",
"rbac_enabled" : true,
"sfs30_url" : "*****",
"sfs_turbo_url" : "*****",
"sfs_url" : "*****",
"supportHcs" : false,
"swr_addr" : "*****",
"swr_user" : "*****"
},
},
"custom" : {
"cluster_id" : "*****",
"cluster_version" : "v1.25.3-r0",
"csi_attacher_detach_worker_threads" : "60",
"csi_attacher_worker_threads" : "60",
"default_vpc_id" : "*****",
"disable_auto_mount_secret" : false,
"enable_node_attacher" : false,
"flow_control" : { },
"multiAZEnabled" : false,
"over_subscription" : "80",
"project_id" : "*****",
"volume_attaching_flow_ctrl" : "0"
},
},
"flavor" : {
"category" : [ "CCE", "Turbo" ],
"description" : "High available",
"name" : "HA",
"replicas" : 2,
"resources" : [ {
"limitsCpu" : "250m",
"limitsMem" : "1500Mi",
"name" : "everest-csi-controller",
"requestsCpu" : "250m",
"requestsMem" : "600Mi"
}, {
"limitsCpu" : "500m",
"limitsMem" : "300Mi",
"name" : "everest-csi-driver",
"requestsCpu" : "100m",
"requestsMem" : "300Mi"
}
]
},
},

```

```

"multiAZPreferred" : {
  "podAntiAffinity" : {
    "preferredDuringSchedulingIgnoredDuringExecution" : [ {
      "podAffinityTerm" : {
        "labelSelector" : {
          "matchExpressions" : [ {
            "key" : "app",
            "operator" : "In",
            "values" : [ "everest-csi-controller" ]
          } ]
        },
        "topologyKey" : "topology.kubernetes.io/zone"
      },
      "weight" : 100
    } ]
  }
},
"multiAZRequired" : {
  "podAntiAffinity" : {
    "requiredDuringSchedulingIgnoredDuringExecution" : [ {
      "labelSelector" : {
        "matchExpressions" : [ {
          "key" : "app",
          "operator" : "In",
          "values" : [ "everest-csi-controller" ]
        } ]
      },
      "topologyKey" : "topology.kubernetes.io/zone"
    } ]
  }
},
"systemAutoInject" : {
  "cluster" : {
    "category" : "CCE",
    "clusterID" : "*****",
    "clusterNetworkMode" : "vpc-router",
    "clusterVersion" : "v1.25.3-r0"
  },
  "user" : {
    "projectID" : "*****"
  }
},
"tolerations" : [ {
  "effect" : "NoExecute",
  "key" : "node.kubernetes.io/not-ready",
  "operator" : "Exists",
  "tolerationSeconds" : 60
}, {
  "effect" : "NoExecute",
  "key" : "node.kubernetes.io/unreachable",
  "operator" : "Exists",
  "tolerationSeconds" : 60
} ]
},
"status" : {
  "status" : "rollbacking",
  "Reason" : "Rollback to 4",
  "message" : "",
  "targetVersions" : [ "2.1.18", "2.1.19" ],
  "isRollbackable" : false,
  "previousVersion" : "2.1.19",
  "currentVersion" : {
    "version" : "2.1.16",
    "input" : {
      "basic" : {
        "bms_url" : "*****",
        "driver_init_image_version" : "2.1.16",
        "ecsEndpoint" : "*****",

```

```

"everest_image_version" : "2.1.16",
"evs_url" : "*****",
"iam_url" : "*****",
"ims_url" : "*****",
"obs_url" : "*****",
"platform" : "linux-amd64",
"sfs30_url" : "*****",
"sfs_turbo_url" : "*****",
"sfs_url" : "*****",
"supportHcs" : false,
"swr_addr" : "*****",
"swr_user" : "*****"
},
"parameters" : {
  "common" : {
    "defaultVPCId" : 0
  },
  "custom" : {
    "cluster_id" : "",
    "csi_attacher_detach_worker_threads" : "60",
    "csi_attacher_worker_threads" : "60",
    "default_vpc_id" : "",
    "disable_auto_mount_secret" : false,
    "enable_node_attacher" : false,
    "flow_control" : { },
    "multiAZEnabled" : false,
    "over_subscription" : "80",
    "project_id" : "",
    "volume_attaching_flow_ctrl" : "0"
  },
  "flavor1" : {
    "description" : "High available",
    "name" : "HA",
    "replicas" : 2,
    "resources" : [ {
      "limitsCpu" : "250m",
      "limitsMem" : "1500Mi",
      "name" : "everest-csi-controller",
      "requestsCpu" : "250m",
      "requestsMem" : "600Mi"
    }, {
      "limitsCpu" : "500m",
      "limitsMem" : "300Mi",
      "name" : "everest-csi-driver",
      "requestsCpu" : "100m",
      "requestsMem" : "300Mi"
    } ]
  },
  "flavor2" : {
    "description" : "Has only one instance",
    "name" : "Single",
    "replicas" : 1,
    "resources" : [ {
      "limitsCpu" : "250m",
      "limitsMem" : "600Mi",
      "name" : "everest-csi-controller",
      "requestsCpu" : "250m",
      "requestsMem" : "600Mi"
    }, {
      "limitsCpu" : "100m",
      "limitsMem" : "300Mi",
      "name" : "everest-csi-driver",
      "requestsCpu" : "100m",
      "requestsMem" : "300Mi"
    } ]
  },
  "flavor3" : {
    "description" : "custom resources",
    "name" : "custom-resources",

```

```
"replicas" : 2,
"resources" : [ {
  "limitsCpu" : "250m",
  "limitsMem" : "2000Mi",
  "name" : "everest-csi-controller",
  "requestsCpu" : "250m",
  "requestsMem" : "1500Mi"
}, {
  "limitsCpu" : "500m",
  "limitsMem" : "300Mi",
  "name" : "everest-csi-driver",
  "requestsCpu" : "100m",
  "requestsMem" : "300Mi"
} ]
}
},
"stable" : true,
"translate" : {
  "en_US" : {
    "addon" : {
      "changeLog" : "*****",
      "description" : "*****"
    },
    "description" : {
      "Parameters.flavor1.description" : "*****",
      "Parameters.flavor1.name" : "*****",
      "Parameters.flavor2.description" : "*****",
      "Parameters.flavor2.name" : "*****",
      "Parameters.flavor3.name" : "*****"
    }
  },
  "zh_CN" : {
    "addon" : {
      "changeLog" : "*****",
      "description" : "*****"
    },
    "description" : {
      "Parameters.flavor1.description" : "*****",
      "Parameters.flavor1.name" : "*****",
      "Parameters.flavor2.description" : "*****",
      "Parameters.flavor2.name" : "*****",
      "Parameters.flavor3.description" : "*****",
      "Parameters.flavor3.name" : "*****"
    }
  }
},
"supportVersions" : null,
"creationTimestamp" : "2023-02-21T16:29:02Z",
"updateTimestamp" : "2023-02-22T06:49:50Z"
}
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
```

```
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class RollbackAddonInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        RollbackAddonInstanceRequest request = new RollbackAddonInstanceRequest();
        request.withId("{id}");
        AddonInstanceRollbackRequest body = new AddonInstanceRollbackRequest();
        body.withClusterID("*****");
        request.withBody(body);
        try {
            RollbackAddonInstanceResponse response = client.rollbackAddonInstance(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()
```

```
try:
    request = RollbackAddonInstanceRequest()
    request.id = "{id}"
    request.body = AddonInstanceRollbackRequest(
        cluster_id="*****"
    )
    response = client.rollback_addon_instance(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RollbackAddonInstanceRequest{
        request.Id = "{id}"
        request.Body = &model.AddonInstanceRollbackRequest{
            ClusterID: "*****",
        }
    }
    response, err := client.RollbackAddonInstance(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Add-on instance version is rolled back.

Error Codes

See [Error Codes](#).

4.6.5 Deleting an Add-on Instance

Function

This API is used to delete an add-on instance.

Calling Method

For details, see [Calling APIs](#).

URI

DELETE /api/v3/addons/{id}

Table 4-738 Path Parameters

Parameter	Mandatory	Type	Description
id	Yes	String	Add-on instance ID.

Table 4-739 Query Parameters

Parameter	Mandatory	Type	Description
cluster_id	No	String	Cluster ID (deprecated). For details about how to obtain the cluster ID, see How Do I Obtain Parameters in the API URI .

Request Parameters

Table 4-740 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-741 Response body parameters

Parameter	Type	Description
-	String	

Example Requests

None

Example Responses

Status code: 200

OK

success

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
```



```
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class DeleteAddonInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteAddonInstanceRequest request = new DeleteAddonInstanceRequest();
        request.withId("{id}");
        try {
            DeleteAddonInstanceResponse response = client.deleteAddonInstance(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteAddonInstanceRequest()
```

```

request.id = "{id}"
response = client.delete_addon_instance(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)

```

Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteAddonInstanceRequest{}
    request.Id = "{id}"
    response, err := client.DeleteAddonInstance(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	OK

Error Codes

See [Error Codes](#).

4.6.6 Querying an Add-on Instance

Function

This API is used to obtain details about an add-on instance.

Calling Method

For details, see [Calling APIs](#).

URI

GET /api/v3/addons/{id}

Table 4-742 Path Parameters

Parameter	Mandatory	Type	Description
id	Yes	String	Add-on instance ID.

Table 4-743 Query Parameters

Parameter	Mandatory	Type	Description
cluster_id	No	String	Cluster ID (deprecated). For details about how to obtain the cluster ID, see How Do I Obtain Parameters in the API URI .

Request Parameters

Table 4-744 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-745 Response body parameters

Parameter	Type	Description
kind	String	API type. The value is fixed at Addon and cannot be changed.
apiVersion	String	API version. The value is fixed at v3 and cannot be changed.
metadata	AddonMetadata object	Basic information about the object. Metadata is a collection of attributes.
spec	InstanceSpec object	Detailed description of the add-on instance.
status	AddonInstanceStatus object	Add-on instance status.

Table 4-746 AddonMetadata

Parameter	Type	Description
uid	String	Unique ID
name	String	Add-on name
alias	String	Add-on alias
labels	Map<String,String>	Add-on labels in key-value pairs. This is a reserved field and does not take effect.

Parameter	Type	Description
annotations	Map<String,String>	Add-on annotations in the format of key-value pairs. <ul style="list-style-type: none"> For add-on installation, the value is fixed at {"addon.install/type":"install"}. For add-on upgrade, the value is fixed at {"addon.upgrade/type":"upgrade"}.
updateTimestamp	String	Updated at
creationTimestamp	String	Created at

Table 4-747 InstanceSpec

Parameter	Type	Description
clusterID	String	Cluster ID.
version	String	Add-on template version, for example, 1.0.0 .
addonTemplateName	String	Add-on template name, for example, coredns .
addonTemplateType	String	Add-on template type.
addonTemplateLogo	String	URL for obtaining the add-on template logo.
addonTemplateLabels	Array of strings	Labels of the add-on template.
description	String	Add-on template description.
values	Map<String,Object>	Add-on template installation parameters (varying depending on the add-on). Set the parameters accordingly.

Table 4-748 AddonInstanceStatus

Parameter	Type	Description
status	String	<p>Add-on instance status. Options:</p> <ul style="list-style-type: none"> ● running: All of the add-on instances are running. This specifies that the add-on runs properly. ● abnormal: The add-on instances are abnormal and the add-on cannot be used. You can click the add-on name to view exceptions. ● installing: The add-on is being installed. ● installFailed: Installing the add-on failed. In this case, uninstall the add-on and then reinstall it. ● upgrading: The add-on is being upgraded. ● upgradeFailed: Upgrading the add-on failed. In this case, upgrade the add-on again or uninstall the add-on and reinstall it. ● deleting: The add-on is being deleted. ● deleteFailed: Deleting the add-on failed. In this case, uninstall the add-on again. ● deleteSuccess: Deleting the add-on succeeded. ● available: Only some instances of the add-on are running. This specifies that some functions of the add-on are available. ● rollbacking: The add-on is being rolled back. ● rollbackFailed: Rolling back the add-on failed. In this case, roll back the add-on again or uninstall the add-on and reinstall it. ● unknown: The add-on chart instance does not exist.
Reason	String	Cause of the add-on installation failure.
message	String	Installation error details.
targetVersions	Array of strings	Versions to which the current add-on version can be upgraded.
currentVersion	Versions object	Information about the current add-on version.
isRollbackable	Boolean	Whether the add-on version can be rolled back to the source version.

Parameter	Type	Description
previousVersion	String	The add-on version before upgrade or rollback

Table 4-749 Versions

Parameter	Type	Description
version	String	Add-on version.
input	Object	Add-on installation parameters.
stable	Boolean	Whether the add-on version is a stable release.
translate	Object	Translation information used by the GUI.
supportVersions	Array of SupportVersions objects	Cluster versions that support the add-on.
creationTimestamp	String	Creation time.
updateTimestamp	String	Update time.

Table 4-750 SupportVersions

Parameter	Type	Description
clusterType	String	Cluster type that supports the add-on.
clusterVersion	Array of strings	Cluster versions that support the add-on. The value is a regular expression.

Example Requests

None

Example Responses

Status code: 200

OK

```
{
  "kind": "Addon",
  "apiVersion": "v3",
  "metadata": {
    "uid": "24b23108-55c0-11e9-926f-0255ac101a31",
    "name": "gpu-beta",
    "alias": "gpu",
```

```
"creationTimestamp" : "2019-04-03T03:25:34Z",
"updateTimestamp" : "2019-04-03T03:25:34Z"
},
"spec" : {
  "clusterID" : "0c0e4a63-5539-11e9-95f7-0255ac10177e",
  "version" : "1.0.0",
  "addonTemplateName" : "gpu-beta",
  "addonTemplateType" : "helm",
  "addonTemplateLogo" : "",
  "addonTemplateLabels" : [ "Accelerator" ],
  "description" : "A device plugin for nvidia.com/gpu resource on nvidia driver",
  "values" : {
    "basic" : {
      "rbac_enabled" : true,
      "swr_addr" : "100.125.6.246:20202",
      "swr_user" : "hwofficial"
    }
  }
},
"status" : {
  "status" : "installing",
  "Reason" : "",
  "message" : "",
  "targetVersions" : null,
  "currentVersion" : {
    "version" : "1.0.0",
    "input" : {
      "basic" : {
        "swr_addr" : "100.125.6.246:20202",
        "swr_user" : "hwofficial"
      }
    },
    "parameters" : {}
  },
  "stable" : true,
  "translate" : {
    "en_US" : {
      "addon" : {
        "changeLog" : "A device plugin for nvidia.com/gpu resource on nvidia driver",
        "description" : "A device plugin for nvidia.com/gpu resource on nvidia driver"
      }
    },
    "zh_CN" : {
      "addon" : {
        "changeLog" : "",
        "description" : ""
      }
    }
  },
  "supportVersions" : null,
  "creationTimestamp" : "2018-10-23T13:14:55Z",
  "updateTimestamp" : "2018-12-07T09:40:24Z"
}
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
```



```
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ShowAddonInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowAddonInstanceRequest request = new ShowAddonInstanceRequest();
        request.withId("{id}");
        try {
            ShowAddonInstanceResponse response = client.showAddonInstance(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
```

```

request = ShowAddonInstanceRequest()
request.id = "{id}"
response = client.show_addon_instance(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)

```

Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowAddonInstanceRequest{}
    request.Id = "{id}"
    response, err := client.ShowAddonInstance(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	OK

Error Codes

See [Error Codes](#).

4.6.7 Listing Add-on Instances

Function

This API is used to list all add-on instances in the cluster.

Calling Method

For details, see [Calling APIs](#).

URI

GET /api/v3/addons

Table 4-751 Query Parameters

Parameter	Mandatory	Type	Description
cluster_id	Yes	String	Cluster ID. For details about how to obtain a cluster ID, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-752 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-753 Response body parameters

Parameter	Type	Description
kind	String	API type. The value is fixed at Addon and cannot be changed.
apiVersion	String	API version. The value is fixed at v3 and cannot be changed.
items	Array of AddonInstance objects	Add-on instance list.

Table 4-754 AddonInstance

Parameter	Type	Description
kind	String	API type. The value is fixed at Addon and cannot be changed.
apiVersion	String	API version. The value is fixed at v3 and cannot be changed.
metadata	AddonMetadata object	Basic information about the object. Metadata is a collection of attributes.
spec	InstanceSpec object	Detailed description of the add-on instance.
status	AddonInstanceStatus object	Add-on instance status.

Table 4-755 AddonMetadata

Parameter	Type	Description
uid	String	Unique ID
name	String	Add-on name
alias	String	Add-on alias
labels	Map<String,String>	Add-on labels in key-value pairs. This is a reserved field and does not take effect.

Parameter	Type	Description
annotations	Map<String,String>	Add-on annotations in the format of key-value pairs. <ul style="list-style-type: none"> For add-on installation, the value is fixed at {"addon.install/type":"install"}. For add-on upgrade, the value is fixed at {"addon.upgrade/type":"upgrade"}.
updateTimestamp	String	Updated at
creationTimestamp	String	Created at

Table 4-756 InstanceSpec

Parameter	Type	Description
clusterID	String	Cluster ID.
version	String	Add-on template version, for example, 1.0.0 .
addonTemplateName	String	Add-on template name, for example, coredns .
addonTemplateType	String	Add-on template type.
addonTemplateLogo	String	URL for obtaining the add-on template logo.
addonTemplateLabels	Array of strings	Labels of the add-on template.
description	String	Add-on template description.
values	Map<String,Object>	Add-on template installation parameters (varying depending on the add-on). Set the parameters accordingly.

Table 4-757 AddonInstanceStatus

Parameter	Type	Description
status	String	<p>Add-on instance status. Options:</p> <ul style="list-style-type: none"> ● running: All of the add-on instances are running. This specifies that the add-on runs properly. ● abnormal: The add-on instances are abnormal and the add-on cannot be used. You can click the add-on name to view exceptions. ● installing: The add-on is being installed. ● installFailed: Installing the add-on failed. In this case, uninstall the add-on and then reinstall it. ● upgrading: The add-on is being upgraded. ● upgradeFailed: Upgrading the add-on failed. In this case, upgrade the add-on again or uninstall the add-on and reinstall it. ● deleting: The add-on is being deleted. ● deleteFailed: Deleting the add-on failed. In this case, uninstall the add-on again. ● deleteSuccess: Deleting the add-on succeeded. ● available: Only some instances of the add-on are running. This specifies that some functions of the add-on are available. ● rollbacking: The add-on is being rolled back. ● rollbackFailed: Rolling back the add-on failed. In this case, roll back the add-on again or uninstall the add-on and reinstall it. ● unknown: The add-on chart instance does not exist.
Reason	String	Cause of the add-on installation failure.
message	String	Installation error details.
targetVersions	Array of strings	Versions to which the current add-on version can be upgraded.
currentVersion	Versions object	Information about the current add-on version.
isRollbackable	Boolean	Whether the add-on version can be rolled back to the source version.

Parameter	Type	Description
previousVersion	String	The add-on version before upgrade or rollback

Table 4-758 Versions

Parameter	Type	Description
version	String	Add-on version.
input	Object	Add-on installation parameters.
stable	Boolean	Whether the add-on version is a stable release.
translate	Object	Translation information used by the GUI.
supportVersions	Array of SupportVersions objects	Cluster versions that support the add-on.
creationTimestamp	String	Creation time.
updateTimestamp	String	Update time.

Table 4-759 SupportVersions

Parameter	Type	Description
clusterType	String	Cluster type that supports the add-on.
clusterVersion	Array of strings	Cluster versions that support the add-on. The value is a regular expression.

Example Requests

None

Example Responses

Status code: 200

ok

```
{
  "kind": "Addon",
  "apiVersion": "v3",
  "items": [ {
    "kind": "Addon",
    "apiVersion": "v3",
    "metadata": {
```

```
"uid" : "8ca259cc-553b-11e9-926f-0255ac101a31",
"name" : "storage-driver",
"alias" : "storage-driver",
"creationTimestamp" : "2019-04-02T11:36:26Z",
"updateTimestamp" : "2019-04-02T11:36:26Z"
},
"spec" : {
  "clusterID" : "0c0e4a63-5539-11e9-95f7-0255ac10177e",
  "version" : "1.0.10",
  "addonTemplateName" : "storage-driver",
  "addonTemplateType" : "helm",
  "addonTemplateLogo" : "https://192.149.48.66/cce-addon-southchina-aw1hz2u/storage-driverlogo.svg",
  "addonTemplateLabels" : [ "Storage" ],
  "description" : "A kubernetes FlexVolume Driver used to support cloud storage",
  "values" : {
    "basic" : {
      "addon_version" : "1.0.10",
      "euleros_version" : "2.2.5",
      "obs_url" : "",
      "platform" : "linux-amd64",
      "swr_addr" : "100.125.6.246:20202",
      "swr_user" : "hwofficial"
    },
    "flavor" : {
      "replicas" : 1
    },
    "parameters" : {}
  }
},
"status" : {
  "status" : "running",
  "Reason" : "Install complete",
  "message" : "",
  "targetVersions" : null,
  "currentVersion" : {
    "version" : "1.0.10",
    "input" : {
      "basic" : {
        "euleros_version" : "2.2.5",
        "obs_url" : "",
        "swr_addr" : "100.125.6.246:20202",
        "swr_user" : "hwofficial"
      },
      "parameters" : {}
    },
    "stable" : true,
    "translate" : {
      "en_US" : {
        "addon" : {
          "changeLog" : "The plug-in is upgraded to enhance the storage plug-in function.",
          "description" : "A kubernetes FlexVolume Driver used to support cloud storage"
        }
      },
      "zh_CN" : {
        "addon" : {
          "changeLog" : "",
          "description" : ""
        }
      }
    },
    "supportVersions" : null,
    "creationTimestamp" : "2019-03-29T13:45:37Z",
    "updateTimestamp" : "2019-03-29T13:45:37Z"
  }
}
}]
}
```


SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ListAddonInstancesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        ListAddonInstancesRequest request = new ListAddonInstancesRequest();
        try {
            ListAddonInstancesResponse response = client.listAddonInstances(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
```

```
# In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = BasicCredentials(ak, sk) \

client = CceClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(CceRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ListAddonInstancesRequest()
    response = client.list_addon_instances(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAddonInstancesRequest{}
    response, err := client.ListAddonInstances(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	ok

Error Codes

See [Error Codes](#).

4.7 Cluster Upgrade

4.7.1 Upgrading a Cluster

Function

Cluster upgrade

 **NOTE**

- Cluster upgrade involves many operations on components. You are advised to upgrade your cluster on the CCE console, a more interactive and intuitive way to reduce operational risks.
- Currently, cluster upgrade APIs are available only upon request.

Calling Method

For details, see [Calling APIs](#).

URI

POST /api/v3/projects/{project_id}/clusters/{cluster_id}/operation/upgrade

Table 4-760 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-761 Request body parameters

Parameter	Mandatory	Type	Description
metadata	Yes	UpgradeClusterRequestMetadata object	Upgrade metadata.
spec	Yes	UpgradeSpec object	Upgrade settings.

Table 4-762 UpgradeClusterRequestMetadata

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	API version. Defaults to v3 .
kind	Yes	String	Resource type. Defaults to UpgradeTask .

Table 4-763 UpgradeSpec

Parameter	Mandatory	Type	Description
clusterUpgradeAction	No	ClusterUpgradeAction object	Cluster upgrade settings.

Table 4-764 ClusterUpgradeAction

Parameter	Mandatory	Type	Description
addons	No	Array of UpgradeAddonConfig objects	Add-on configuration list.
nodeOrder	No	Map<String,Array< NodePriority >>	Upgrade sequence of nodes in the node pool. NOTE key indicates the node pool ID. Defaults to DefaultPool .

Parameter	Mandatory	Type	Description
nodePoolOrder	No	Map<String,Integer>	Upgrade sequence of a node pool, in key-value pairs. NOTE key indicates the node pool ID. Defaults to DefaultPool . value means the priority of a node pool. Defaults to 0 , indicating the lowest priority. A larger value indicates a higher priority.
strategy	Yes	UpgradeStrategy object	Upgrade policy.
targetVersion	Yes	String	Target cluster version, for example, v1.23.

Table 4-765 UpgradeAddonConfig

Parameter	Mandatory	Type	Description
addonTemplateName	Yes	String	Add-on name.
operation	Yes	String	Execution action. For current upgrades, the value can be patch .
version	Yes	String	Target add-on version. NOTE The target add-on version must match the target cluster version. For details about the mapping between the add-on versions and cluster versions, see Listing Add-on Templates .
values	No	Object	Add-on parameter list, in key-value pairs.

Table 4-766 NodePriority

Parameter	Mandatory	Type	Description
nodeSelector	Yes	NodeSelector object	Node label selector, which selects a batch of nodes.

Parameter	Mandatory	Type	Description
priority	Yes	Integer	Priority of this batch of nodes. Defaults to 0 , indicating the lowest priority. A larger value indicates a higher priority.

Table 4-767 NodeSelector

Parameter	Mandatory	Type	Description
key	Yes	String	Label key.
value	No	Array of strings	Label value list.
operator	Yes	String	Logical operators of labels.

Table 4-768 UpgradeStrategy

Parameter	Mandatory	Type	Description
type	Yes	String	Upgrade policy type. Supports only <code>inPlaceRollingUpdate</code> .
<code>inPlaceRollingUpdate</code>	No	InPlaceRollingUpdate object	In-place upgrade settings. Mandatory when the in-place upgrade policy type is specified.

Table 4-769 InPlaceRollingUpdate

Parameter	Mandatory	Type	Description
<code>userDefinedStep</code>	No	Integer	Node upgrade step, from 1 to 40. Recommended: 20.

Response Parameters

Status code: 200

Table 4-770 Response body parameters

Parameter	Type	Description
metadata	UpgradeClusterResponseMetadata object	Upgrade task metadata
spec	UpgradeResponseSpec object	Upgrade settings

Table 4-771 UpgradeClusterResponseMetadata

Parameter	Type	Description
uid	String	Upgrade task ID. You can obtain the progress by calling the API for obtaining cluster upgrade task details.

Table 4-772 UpgradeResponseSpec

Parameter	Type	Description
clusterUpgradeAction	ClusterUpgradeResponseAction object	Cluster upgrade settings

Table 4-773 ClusterUpgradeResponseAction

Parameter	Type	Description
version	String	Current cluster version
targetVersion	String	Target cluster version, for example, v1.23.
targetPlatformVersion	String	Platform version of the target cluster, which is an internal version of the cluster version and cannot be specified.
strategy	UpgradeStrategy object	Upgrade policies
config	Object	Cluster configuration specified during an upgrade

Table 4-774 UpgradeStrategy

Parameter	Type	Description
type	String	Upgrade policy type. Supports only inPlaceRollingUpdate.
inPlaceRollingUpdate	InPlaceRollingUpdate object	In-place upgrade settings. Mandatory when the in-place upgrade policy type is specified.

Table 4-775 InPlaceRollingUpdate

Parameter	Type	Description
userDefinedStep	Integer	Node upgrade step, from 1 to 40. Recommended: 20.

Example Requests

Upgrade the cluster to v1.23 and set the node upgrade step to 20.

```
POST /api/v3/projects/{project_id}/clusters/{cluster_id}/operation/upgrade
```

```
{
  "metadata": {
    "apiVersion": "v3",
    "kind": "UpgradeTask"
  },
  "spec": {
    "clusterUpgradeAction": {
      "strategy": {
        "type": "inPlaceRollingUpdate",
        "inPlaceRollingUpdate": {
          "userDefinedStep": 20
        }
      }
    },
    "targetVersion": "v1.23"
  }
}
```

Example Responses

Status code: 200

Cluster upgrade requested.

```
{
  "metadata": {
    "uid": "976a33e2-f545-11ed-87af-0255ac1002c2"
  },
  "spec": {
    "clusterUpgradeAction": {
      "version": "v1.19.16-r20",
      "targetVersion": "v1.23.8-r0",
      "targetPlatformVersion": "cce.10",
      "strategy": {
        "type": "inPlaceRollingUpdate",

```



```
"inPlaceRollingUpdate" : {  
  "userDefinedStep" : 20  
},  
"config" : {}  
}  
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Upgrade the cluster to v1.23 and set the node upgrade step to 20.

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cce.v3.region.CceRegion;  
import com.huaweicloud.sdk.cce.v3.*;  
import com.huaweicloud.sdk.cce.v3.model.*;  
  
public class UpgradeClusterSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        CceClient client = CceClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))  
            .build();  
        UpgradeClusterRequest request = new UpgradeClusterRequest();  
        request.withClusterId("{cluster_id}");  
        UpgradeClusterRequestBody body = new UpgradeClusterRequestBody();  
        InPlaceRollingUpdate inPlaceRollingUpdateStrategy = new InPlaceRollingUpdate();  
        inPlaceRollingUpdateStrategy.withUserDefinedStep(20);  
        UpgradeStrategy strategyClusterUpgradeAction = new UpgradeStrategy();  
        strategyClusterUpgradeAction.withType("inPlaceRollingUpdate")  
            .withInPlaceRollingUpdate(inPlaceRollingUpdateStrategy);  
        ClusterUpgradeAction clusterUpgradeActionSpec = new ClusterUpgradeAction();  
        clusterUpgradeActionSpec.withStrategy(strategyClusterUpgradeAction)  
            .withTargetVersion("v1.23");  
        UpgradeSpec specbody = new UpgradeSpec();  
        specbody.withClusterUpgradeAction(clusterUpgradeActionSpec);  
        UpgradeClusterRequestMetadata metadatabody = new UpgradeClusterRequestMetadata();  
        metadatabody.withApiVersion("v3")  
            .withKind("UpgradeTask");  
        body.withSpec(specbody);  
    }  
}
```

```
body.withMetadata(metadataBody);
request.withBody(body);
try {
    UpgradeClusterResponse response = client.upgradeCluster(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

Upgrade the cluster to v1.23 and set the node upgrade step to 20.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpgradeClusterRequest()
        request.cluster_id = "{cluster_id}"
        inPlaceRollingUpdateStrategy = InPlaceRollingUpdate(
            user_defined_step=20
        )
        strategyClusterUpgradeAction = UpgradeStrategy(
            type="inPlaceRollingUpdate",
            in_place_rolling_update=inPlaceRollingUpdateStrategy
        )
        clusterUpgradeActionSpec = ClusterUpgradeAction(
            strategy=strategyClusterUpgradeAction,
            target_version="v1.23"
        )
        specbody = UpgradeSpec(
            cluster_upgrade_action=clusterUpgradeActionSpec
        )
        metadataBody = UpgradeClusterRequestMetadata(
            api_version="v3",
            kind="UpgradeTask"
        )
        request.body = UpgradeClusterRequestBody(
```

```
        spec=specbody,
        metadata=metadatabody
    )
    response = client.upgrade_cluster(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

Upgrade the cluster to v1.23 and set the node upgrade step to 20.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpgradeClusterRequest{}
    request.ClusterId = "{cluster_id}"
    userDefinedStepInPlaceRollingUpdate := int32(20)
    inPlaceRollingUpdateStrategy := &model.InPlaceRollingUpdate{
        UserDefinedStep: &userDefinedStepInPlaceRollingUpdate,
    }
    strategyClusterUpgradeAction := &model.UpgradeStrategy{
        Type: "inPlaceRollingUpdate",
        InPlaceRollingUpdate: inPlaceRollingUpdateStrategy,
    }
    clusterUpgradeActionSpec := &model.ClusterUpgradeAction{
        Strategy: strategyClusterUpgradeAction,
        TargetVersion: "v1.23",
    }
    specbody := &model.UpgradeSpec{
        ClusterUpgradeAction: clusterUpgradeActionSpec,
    }
    metadatabody := &model.UpgradeClusterRequestMetadata{
        ApiVersion: "v3",
        Kind: "UpgradeTask",
    }
    request.Body = &model.UpgradeClusterRequestBody{
```

```
Spec: specbody,  
Metadata: metadatabody,  
}  
response, err := client.UpgradeCluster(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Cluster upgrade requested.

Error Codes

See [Error Codes](#).

4.7.2 Obtaining Cluster Upgrade Task Details

Function

Upgrade task ID, obtained from the **uid** field in the response body after the cluster upgrade API is called.

NOTE

- Cluster upgrade involves many operations on components. You are advised to upgrade your cluster on the CCE console, a more interactive and intuitive way to reduce operational risks.
- Currently, cluster upgrade APIs are available only upon request.

Calling Method

For details, see [Calling APIs](#).

URI

GET /api/v3/projects/{project_id}/clusters/{cluster_id}/operation/upgrade/tasks/{task_id}

Table 4-776 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
task_id	Yes	String	Upgrade task ID, obtained from the uid field in the response body after the cluster upgrade API is called.

Request Parameters

None

Response Parameters

Status code: 200

Table 4-777 Response body parameters

Parameter	Type	Description
apiVersion	String	API version. Defaults to v3 .
kind	String	Resource type. Defaults to UpgradeTask .
metadata	UpgradeTask Metadata object	Upgrade task metadata.
spec	UpgradeTask Spec object	Upgrade task information.
status	UpgradeTask Status object	Upgrade task status.

Table 4-778 UpgradeTaskMetadata

Parameter	Type	Description
uid	String	Upgrade task ID.

Parameter	Type	Description
creationTimes tamp	String	Time when a task was created.
updateTimest amp	String	Time when a task is updated.

Table 4-779 UpgradeTaskSpec

Parameter	Type	Description
version	String	Source cluster version.
targetVersion	String	Target cluster version to upgrade to.
items	Object	Additional information about the upgrade task.

Table 4-780 UpgradeTaskStatus

Parameter	Type	Description
phase	String	Indicates the status of the upgrade task. Init: initializing NOTE Queuing: waiting <ul style="list-style-type: none"> • Running: The driver is running. NOTE Pause: paused <ul style="list-style-type: none"> • Success: successful. NOTE Failed: failed
progress	String	Upgrade task progress.
completionTi me	String	End time of the upgrade task.

Example Requests

None

Example Responses

Status code: 200

Cluster upgrade task details obtained.

```
{
  "kind": "UpgradeTask",
```

```
"apiVersion" : "v3",
"metadata" : {
  "uid" : "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "creationTimestamp" : "2022-12-16 13:40:20.75671 +0800 CST",
  "updateTimestamp" : "2022-12-16 13:40:20.756712 +0800 CST"
},
"spec" : {
  "version" : "v1.19.16-r4",
  "targetVersion" : "v1.23.5-r0"
},
"status" : {
  "phase" : "Init",
  "progress" : "0.00",
  "completionTime" : "2022-12-16 13:40:20.756712 +0800 CST"
}
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ShowUpgradeClusterTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowUpgradeClusterTaskRequest request = new ShowUpgradeClusterTaskRequest();
        request.withClusterId("{cluster_id}");
        request.withTaskId("{task_id}");
        try {
            ShowUpgradeClusterTaskResponse response = client.showUpgradeClusterTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
        }
    }
}
```

```
        System.out.println(e.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowUpgradeClusterTaskRequest()
        request.cluster_id = "{cluster_id}"
        request.task_id = "{task_id}"
        response = client.show_upgrade_cluster_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
```



```

WithAk(ak).
WithSk(sk).
WithProjectId(projectId).
Build()

client := cce.NewCceClient(
    cce.CceClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowUpgradeClusterTaskRequest{}
request.ClusterId = "{cluster_id}"
request.TaskId = "{task_id}"
response, err := client.ShowUpgradeClusterTask(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Cluster upgrade task details obtained.

Error Codes

See [Error Codes](#).

4.7.3 Retrying a Cluster Upgrade Task

Function

Execute the failed cluster upgrade task again.

NOTE

- Cluster upgrade involves many operations on components. You are advised to upgrade your cluster on the CCE console, a more interactive and intuitive way to reduce operational risks.
- Currently, cluster upgrade APIs are available only upon request.

Calling Method

For details, see [Calling APIs](#).

URI

POST /api/v3/projects/{project_id}/clusters/{cluster_id}/operation/upgrade/retry

Table 4-781 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

None

Response Parameters

None

Example Requests

None

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class RetryUpgradeClusterTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
```

```
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

CceClient client = CceClient.newBuilder()
    .withCredential(auth)
    .withRegion(CceRegion.valueOf("<YOUR REGION>"))
    .build();
RetryUpgradeClusterTaskRequest request = new RetryUpgradeClusterTaskRequest();
request.withClusterId("{cluster_id}");
try {
    RetryUpgradeClusterTaskResponse response = client.retryUpgradeClusterTask(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RetryUpgradeClusterTaskRequest()
        request.cluster_id = "{cluster_id}"
        response = client.retry_upgrade_cluster_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RetryUpgradeClusterTaskRequest{}
    request.ClusterId = "{cluster_id}"
    response, err := client.RetryUpgradeClusterTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Cluster upgrade retry requested.

Error Codes

See [Error Codes](#).

4.7.4 Suspending a Cluster Upgrade Task (Deprecated)

Function

Suspend the cluster upgrade task.

NOTE

- Cluster upgrade involves many operations on components. You are advised to upgrade your cluster on the CCE console, a more interactive and intuitive way to reduce operational risks.
- Currently, cluster upgrade APIs are available only upon request.

Calling Method

For details, see [Calling APIs](#).

URI

POST /api/v3/projects/{project_id}/clusters/{cluster_id}/operation/upgrade/pause

Table 4-782 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

None

Response Parameters

None

Example Requests

None

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class PauseUpgradeClusterTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        PauseUpgradeClusterTaskRequest request = new PauseUpgradeClusterTaskRequest();
        request.withClusterId("{cluster_id}");
        try {
            PauseUpgradeClusterTaskResponse response = client.pauseUpgradeClusterTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
```

```
# The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
variables and decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId) \

client = CceClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(CceRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = PauseUpgradeClusterTaskRequest()
    request.cluster_id = "{cluster_id}"
    response = client.pause_upgrade_cluster_task(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.PauseUpgradeClusterTaskRequest{}
    request.ClusterId = "{cluster_id}"
    response, err := client.PauseUpgradeClusterTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

```
}  
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Cluster upgrade suspension requested.

Error Codes

See [Error Codes](#).

4.7.5 Continuing to Execute a Cluster Upgrade Task (Deprecated)

Function

Resume the suspended cluster upgrade task.

NOTE

- Cluster upgrade involves many operations on components. You are advised to upgrade your cluster on the CCE console, a more interactive and intuitive way to reduce operational risks.
- Currently, cluster upgrade APIs are available only upon request.

Calling Method

For details, see [Calling APIs](#).

URI

POST /api/v3/projects/{project_id}/clusters/{cluster_id}/operation/upgrade/continue

Table 4-783 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Parameter	Mandatory	Type	Description
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

None

Response Parameters

None

Example Requests

None

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ContinueUpgradeClusterTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);
```

```
CceClient client = CceClient.newBuilder()
    .withCredential(auth)
    .withRegion(CceRegion.valueOf("<YOUR REGION>"))
    .build();
ContinueUpgradeClusterTaskRequest request = new ContinueUpgradeClusterTaskRequest();
request.withClusterId("{cluster_id}");
try {
    ContinueUpgradeClusterTaskResponse response = client.continueUpgradeClusterTask(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ContinueUpgradeClusterTaskRequest()
        request.cluster_id = "{cluster_id}"
        response = client.continue_upgrade_cluster_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
```

```

    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ContinueUpgradeClusterTaskRequest{}
    request.ClusterId = "{cluster_id}"
    response, err := client.ContinueUpgradeClusterTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Cluster upgrade resumption requested.

Error Codes

See [Error Codes](#).

4.7.6 Obtaining a List of Cluster Upgrade Task Details

Function

Obtain a list of cluster upgrade task details.

Calling Method

For details, see [Calling APIs](#).

URI

GET /api/v3/projects/{project_id}/clusters/{cluster_id}/operation/upgrade/tasks

Table 4-784 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

None

Response Parameters

Status code: 200

Table 4-785 Response body parameters

Parameter	Type	Description
apiVersion	String	API version, which defaults to v3
kind	String	Resource type
metadata	UpgradeTask Metadata object	Metadata
items	Array of UpgradeTask ResponseBody objects	Cluster upgrade tasks

Table 4-786 UpgradeTaskResponseBody

Parameter	Type	Description
apiVersion	String	API version. Defaults to v3 .
kind	String	Resource type. Defaults to UpgradeTask .
metadata	UpgradeTask Metadata object	Upgrade task metadata.
spec	UpgradeTask Spec object	Upgrade task information.
status	UpgradeTask Status object	Upgrade task status.

Table 4-787 UpgradeTaskMetadata

Parameter	Type	Description
uid	String	Upgrade task ID.
creationTimestamp	String	Time when a task was created.
updateTimestamp	String	Time when a task is updated.

Table 4-788 UpgradeTaskSpec

Parameter	Type	Description
version	String	Source cluster version.
targetVersion	String	Target cluster version to upgrade to.
items	Object	Additional information about the upgrade task.

Table 4-789 UpgradeTaskStatus

Parameter	Type	Description
phase	String	Indicates the status of the upgrade task. Init: initializing NOTE Queuing: waiting <ul style="list-style-type: none"> ● Running: The driver is running. NOTE Pause: paused <ul style="list-style-type: none"> ● Success: successful. NOTE Failed: failed
progress	String	Upgrade task progress.
completionTime	String	End time of the upgrade task.

Example Requests

None

Example Responses

Status code: 200

The list of cluster upgrade task details is obtained.

```
{
  "kind": "List",
  "apiVersion": "v3",
  "metadata": { },
  "items": [ {
    "kind": "UpgradeTask",
    "apiVersion": "v3",
    "metadata": {
      "uid": "f40cafed-7bf1-4c3b-b619-80113b4bbb18",
      "creationTimestamp": "2023-11-24 16:41:12.09236 +0800 CST",
      "updateTimestamp": "2023-11-24 16:44:05.634206 +0800 CST"
    },
    "spec": {
      "version": "v1.17.17-r0",
      "targetVersion": "v1.19.16-r80"
    },
    "status": {
      "phase": "Success",
      "completionTime": "2023-11-24 16:44:05.634206 +0800 CST"
    }
  }, {
    "kind": "UpgradeTask",
    "apiVersion": "v3",
    "metadata": {
      "uid": "91755b96-5fd8-4a6a-bda1-983de9055996",
      "creationTimestamp": "2023-11-24 19:54:35.194306 +0800 CST",
      "updateTimestamp": "2023-11-24 20:14:35.194306 +0800 CST"
    },
    "spec": {
```

```
"version" : "v1.19.16-r80",
"targetVersion" : "v1.23.8-r10"
},
"status" : {
  "phase" : "Success",
  "completionTime" : "2023-11-24 20:14:35.194306 +0800 CST"
}
}
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ListUpgradeClusterTasksSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        ListUpgradeClusterTasksRequest request = new ListUpgradeClusterTasksRequest();
        request.withClusterId("{cluster_id}");
        try {
            ListUpgradeClusterTasksResponse response = client.listUpgradeClusterTasks(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListUpgradeClusterTasksRequest()
        request.cluster_id = "{cluster_id}"
        response = client.list_upgrade_cluster_tasks(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```



```

Build()

request := &model.ListUpgradeClusterTasksRequest{}
request.ClusterId = "{cluster_id}"
response, err := client.ListUpgradeClusterTasks(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The list of cluster upgrade task details is obtained.

Error Codes

See [Error Codes](#).

4.7.7 Pre-upgrade Check

Function

Pre-upgrade check

Calling Method

For details, see [Calling APIs](#).

URI

POST /api/v3/projects/{project_id}/clusters/{cluster_id}/operation/precheck

Table 4-790 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Parameter	Mandatory	Type	Description
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-791 Request body parameters

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	API version, which defaults to v3
kind	Yes	String	Resource type, which defaults to PreCheckTask
spec	Yes	PrecheckSpec object	spec is an element type of the collection class. The main body of the configuration that needs to be checked before an upgrade is provided in spec . CCE checks the configuration based on the spec description.

Table 4-792 PrecheckSpec

Parameter	Mandatory	Type	Description
clusterID	No	String	Cluster ID
clusterVersion	No	String	Cluster version
targetVersion	No	String	Target Version
skippedCheckItemList	No	Array of skippedCheckItem objects	Skipped check items

Table 4-793 skippedCheckItemList

Parameter	Mandatory	Type	Description
name	No	String	Names of the skipped check items

Parameter	Mandatory	Type	Description
resourceSelector	No	resourceSelector object	Resource tag selector. This parameter is available only for node check, but not for cluster check or add-on check.

Table 4-794 resourceSelector

Parameter	Mandatory	Type	Description
key	Yes	String	Tag key. Options: <ul style="list-style-type: none"> • node.uid: node UID
values	No	Array of strings	Tag values
operator	Yes	String	Logical operator of a tag. Options: <ul style="list-style-type: none"> • In

Response Parameters

Status code: 200

Table 4-795 Response body parameters

Parameter	Type	Description
apiVersion	String	API version
kind	String	Resource type
metadata	PrecheckClusterResponseMetadata object	Pre-upgrade check metadata
spec	PrecheckSpec object	spec is an element type of the collection class. The main body of the configuration that needs to be checked before an upgrade is provided in spec . CCE checks the configuration based on the spec description.
status	PrecheckStatus object	Pre-upgrade check status of a cluster

Table 4-796 PrecheckCluserResponseMetadata

Parameter	Type	Description
uid	String	Check the task ID.

Table 4-797 PrecheckSpec

Parameter	Type	Description
clusterID	String	Cluster ID
clusterVersion	String	Cluster version
targetVersion	String	Target Version
skippedCheckItem temList	Array of skippedCheckItem temList objects	Skipped check items

Table 4-798 skippedCheckItemList

Parameter	Type	Description
name	String	Names of the skipped check items
resourceSelector	resourceSelector object	Resource tag selector. This parameter is available only for node check, but not for cluster check or add-on check.

Table 4-799 resourceSelector

Parameter	Type	Description
key	String	Tag key. Options: <ul style="list-style-type: none"> node.uid: node UID
values	Array of strings	Tag values
operator	String	Logical operator of a tag. Options: <ul style="list-style-type: none"> In

Table 4-800 PrecheckStatus

Parameter	Type	Description
phase	String	Status. Options: <ul style="list-style-type: none"> • Init: initializing • Running • Success • Failed • Error
expireTimeStamp	String	Check result expiration time
message	String	Information, which typically indicates the log for execution errors
clusterCheckStatus	clusterCheckStatus object	Status of a cluster restrictions check
addonCheckStatus	addonCheckStatus object	Status of an add-on check
nodeCheckStatus	nodeCheckStatus object	Node check status

Table 4-801 clusterCheckStatus

Parameter	Type	Description
phase	String	Status. Options: <ul style="list-style-type: none"> • Init: initializing • Running • Success • Failed
itemsStatus	Array of PreCheckItemStatus objects	Compliance set by check item

Table 4-802 addonCheckStatus

Parameter	Type	Description
phase	String	Status. Options: <ul style="list-style-type: none"> • Init: initializing • Running • Success • Failed
itemsStatus	Array of PreCheckItemStatus objects	Compliance set by check item

Table 4-803 nodeCheckStatus

Parameter	Type	Description
phase	String	Status. Options: <ul style="list-style-type: none"> • Init: initializing • Running • Success • Failed
nodeStageStatus	Array of NodeStageStatus objects	Node check status

Table 4-804 NodeStageStatus

Parameter	Type	Description
nodeInfo	NodeInfo object	Node information
itemsStatus	Array of PreCheckItemStatus objects	Compliance set by check item

Table 4-805 NodeInfo

Parameter	Type	Description
uid	String	Node UID

Parameter	Type	Description
name	String	Node name
status	String	Status
nodeType	String	Node type

Table 4-806 PreCheckItemStatus

Parameter	Type	Description
name	String	Check item
kind	String	Check item type. Options: <ul style="list-style-type: none"> ● Exception: exceptions that need to be resolved by you ● Risk: risks that can be skipped after you confirm them
group	String	Check item group. Options: <ul style="list-style-type: none"> ● LimitCheck: cluster limit check ● MasterCheck: master node check ● NodeCheck: worker node check ● AddonCheck: add-on check ● ExecuteException: process error check
level	String	Check item severity. Options: <ul style="list-style-type: none"> ● Info: information ● Warning ● Fatal: critical
phase	String	Status. Options: <ul style="list-style-type: none"> ● Init: initializing ● Running ● Success ● Failed
message	String	Information
riskSource	riskSource object	Risk item
errorCodes	Array of strings	Error code set

Table 4-807 riskSource

Parameter	Type	Description
configurationRisks	Array of configurationRisks objects	Configuration risk
deprecatedAPIRisks	Array of deprecatedAPIRisks objects	Deprecated API risk
nodeRisks	Array of nodeRisks objects	Node risk
addonRisks	Array of addonRisks objects	Add-on risk

Table 4-808 configurationRisks

Parameter	Type	Description
package	String	Component
sourceFile	String	How to Obtain
nodeMsg	String	Node information
field	String	Parameter Value
operation	String	Change an operation type.
originalValue	String	Original Value
value	String	Current Value

Table 4-809 deprecatedAPIRisks

Parameter	Type	Description
url	String	Request path, for example, /apis/policy/v1beta1/podsecuritypolicies
userAgent	String	Client information

Table 4-810 nodeRisks

Parameter	Type	Description
NodeID	String	Worker node ID

Table 4-811 addonRisks

Parameter	Type	Description
addonTemplate	String	Name of the add-on template
alias	String	Add-on alias

Example Requests

Request body for a cluster pre-upgrade check

POST /api/v3/projects/{project_id}/clusters/{cluster_id}/operation/precheck

```
{
  "kind": "PreCheckTask",
  "apiVersion": "v3",
  "spec": {
    "clusterID": "8978deaa-1743-11ee-8e46-0255ac10004c",
    "clusterVersion": "v1.15.11-r1",
    "targetVersion": "v1.19.16-r80",
    "skippedCheckItemList": [ ]
  }
}
```

Example Responses

Status code: 200

Cluster pre-upgrade check succeeded.

```
{
  "kind": "PreCheckTask",
  "apiVersion": "v3",
  "metadata": {
    "uid": "9991b45e-a2be-4b49-aca4-50a25fa6f81e"
  },
  "spec": {
    "clusterID": "8978deaa-1743-11ee-8e46-0255ac10004c",
    "clusterVersion": "v1.15.11-r1",
    "targetVersion": "v1.19.16-r80"
  },
  "status": {
    "phase": "Init",
    "clusterCheckStatus": {
      "phase": "Init"
    },
    "addonCheckStatus": {
      "phase": "Init"
    },
    "nodeCheckStatus": {
      "phase": "Init"
    }
  }
}
```

```
}  
}
```

Status Codes

Status Code	Description
200	Cluster pre-upgrade check succeeded.

Error Codes

See [Error Codes](#).

4.7.8 Obtaining Details About a Pre-upgrade Check Task of a Cluster

Function

Obtain details about a pre-upgrade check task of a cluster. The task ID is obtained from the **uid** field in the response body after the cluster check API is called.

Calling Method

For details, see [Calling APIs](#).

URI

GET /api/v3/projects/{project_id}/clusters/{cluster_id}/operation/precheck/tasks/{task_id}

Table 4-812 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
task_id	Yes	String	Upgrade task ID, obtained from the uid field in the response body after the cluster upgrade API is called.

Request Parameters

None

Response Parameters

Status code: 200

Table 4-813 Response body parameters

Parameter	Type	Description
apiVersion	String	API version, which defaults to v3
kind	String	Resource type, which defaults to PreCheckTask
metadata	PrecheckTaskMetadata object	Metadata of a pre-upgrade check task
spec	PrecheckSpec object	Information of a pre-upgrade check task
status	PrecheckStatus object	Status of a pre-upgrade check task

Table 4-814 PrecheckTaskMetadata

Parameter	Type	Description
uid	String	Task ID
creationTimestamp	String	Time when the task was created
updateTimestamp	String	Time when the task was updated

Table 4-815 PrecheckSpec

Parameter	Type	Description
clusterID	String	Cluster ID
clusterVersion	String	Cluster version
targetVersion	String	Target Version
skippedCheckItemList	Array of skippedCheckItem objects	Skipped check items

Table 4-816 skippedCheckItemList

Parameter	Type	Description
name	String	Names of the skipped check items
resourceSelector or	resourceSelector object	Resource tag selector. This parameter is available only for node check, but not for cluster check or add-on check.

Table 4-817 resourceSelector

Parameter	Type	Description
key	String	Tag key. Options: <ul style="list-style-type: none"> • node.uid: node UID
values	Array of strings	Tag values
operator	String	Logical operator of a tag. Options: <ul style="list-style-type: none"> • In

Table 4-818 PrecheckStatus

Parameter	Type	Description
phase	String	Status. Options: <ul style="list-style-type: none"> • Init: initializing • Running • Success • Failed • Error
expireTimeStamp	String	Check result expiration time
message	String	Information, which typically indicates the log for execution errors
clusterCheckStatus	clusterCheckStatus object	Status of a cluster restrictions check
addonCheckStatus	addonCheckStatus object	Status of an add-on check
nodeCheckStatus	nodeCheckStatus object	Node check status

Table 4-819 clusterCheckStatus

Parameter	Type	Description
phase	String	Status. Options: <ul style="list-style-type: none"> • Init: initializing • Running • Success • Failed
itemsStatus	Array of PreCheckItemStatus objects	Compliance set by check item

Table 4-820 addonCheckStatus

Parameter	Type	Description
phase	String	Status. Options: <ul style="list-style-type: none"> • Init: initializing • Running • Success • Failed
itemsStatus	Array of PreCheckItemStatus objects	Compliance set by check item

Table 4-821 nodeCheckStatus

Parameter	Type	Description
phase	String	Status. Options: <ul style="list-style-type: none"> • Init: initializing • Running • Success • Failed
nodeStageStatus	Array of NodeStageStatus objects	Node check status

Table 4-822 NodeStageStatus

Parameter	Type	Description
nodeInfo	NodeInfo object	Node information
itemsStatus	Array of PreCheckItemStatus objects	Compliance set by check item

Table 4-823 NodeInfo

Parameter	Type	Description
uid	String	Node UID
name	String	Node name
status	String	Status
nodeType	String	Node type

Table 4-824 PreCheckItemStatus

Parameter	Type	Description
name	String	Check item
kind	String	Check item type. Options: <ul style="list-style-type: none"> ● Exception: exceptions that need to be resolved by you ● Risk: risks that can be skipped after you confirm them
group	String	Check item group. Options: <ul style="list-style-type: none"> ● LimitCheck: cluster limit check ● MasterCheck: master node check ● NodeCheck: worker node check ● AddonCheck: add-on check ● ExecuteException: process error check
level	String	Check item severity. Options: <ul style="list-style-type: none"> ● Info: information ● Warning ● Fatal: critical

Parameter	Type	Description
phase	String	Status. Options: <ul style="list-style-type: none"> • Init: initializing • Running • Success • Failed
message	String	Information
riskSource	riskSource object	Risk item
errorCodes	Array of strings	Error code set

Table 4-825 riskSource

Parameter	Type	Description
configuration Risks	Array of configuration Risks objects	Configuration risk
deprecatedAPI Risks	Array of deprecatedAPI Risks objects	Deprecated API risk
nodeRisks	Array of nodeRisks objects	Node risk
addonRisks	Array of addonRisks objects	Add-on risk

Table 4-826 configurationRisks

Parameter	Type	Description
package	String	Component
sourceFile	String	How to Obtain
nodeMsg	String	Node information
field	String	Parameter Value
operation	String	Change an operation type.

Parameter	Type	Description
originalValue	String	Original Value
value	String	Current Value

Table 4-827 deprecatedAPIRisks

Parameter	Type	Description
url	String	Request path, for example, /apis/policy/v1beta1/podsecuritypolicies
userAgent	String	Client information

Table 4-828 nodeRisks

Parameter	Type	Description
NodeID	String	Worker node ID

Table 4-829 addonRisks

Parameter	Type	Description
addonTemplate	String	Name of the add-on template
alias	String	Add-on alias

Example Requests

None

Example Responses

Status code: 200

The details about a pre-upgrade check task of a cluster are obtained.

```
{
  "kind": "PreCheckTask",
  "apiVersion": "v3",
  "metadata": {
    "uid": "f61e008c-1600-41c0-9bde-121de5a30660",
    "creationTimestamp": "2023-11-25 07:20:04.592972 +0000 UTC",
    "updateTimestamp": "2023-11-25 07:21:05.518966 +0000 UTC"
  },
  "spec": {
    "clusterVersion": "v1.19.16-r4",
    "targetVersion": "v1.23.5-r0"
  }
}
```



```

},
"status": {
  "phase": "Success",
  "expireTimeStamp": "2023-11-25 08:21:05.518966 +0000 UTC",
  "clusterCheckStatus": {
    "phase": "Success",
    "itemsStatus": [ {
      "name": "DeprecatedApiCheck",
      "kind": "Risk",
      "group": "LimitCheck",
      "level": "Info",
      "phase": "Success",
      "message": "check item succeed",
      "riskSource": { }
    }, {
      "name": "NodeContainerdPodRestartRisk",
      "kind": "Risk",
      "group": "LimitCheck",
      "level": "Warning",
      "phase": "Success",
      "message": "check item succeed",
      "riskSource": { }
    }, {
      "name": "ResiduePackageVersion",
      "kind": "Exception",
      "group": "LimitCheck",
      "level": "Fatal",
      "phase": "Success",
      "message": "check item succeed",
      "riskSource": { }
    }
  ]
},
"addonCheckStatus": {
  "phase": "Success",
  "itemsStatus": [ {
    "name": "AddonLimit",
    "kind": "Exception",
    "group": "AddonCheck",
    "level": "Warning",
    "phase": "Success",
    "message": "check item succeed",
    "riskSource": { }
  }, {
    "name": "CoreDNSConfLimit",
    "kind": "Exception",
    "group": "AddonCheck",
    "level": "Fatal",
    "phase": "Success",
    "message": "check item succeed",
    "riskSource": { }
  }
]
},
"nodeCheckStatus": {
  "phase": "Success"
}
}
}

```

Status Codes

Status Code	Description
200	The details about a pre-upgrade check task of a cluster are obtained.

Error Codes

See [Error Codes](#).

4.7.9 Obtaining a List of Pre-upgrade Check Tasks of a Cluster

Function

Obtain a list of pre-upgrade check tasks of a cluster.

Calling Method

For details, see [Calling APIs](#).

URI

GET /api/v3/projects/{project_id}/clusters/{cluster_id}/operation/precheck/tasks

Table 4-830 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

None

Response Parameters

Status code: 200

Table 4-831 Response body parameters

Parameter	Type	Description
apiVersion	String	API version, which defaults to v3
kind	String	Type
metadata	Metadata object	Basic information about the object. Metadata is a collection of attributes.

Parameter	Type	Description
items	Array of PrecheckClusterTask objects	Cluster check tasks

Table 4-832 Metadata

Parameter	Type	Description
uid	String	Unique ID.
name	String	Resource name
labels	Map<String,String>	Resource labels in key-value pairs. This is a reserved field and does not take effect.
annotations	Map<String,String>	Resource annotations in the format of key-value pairs.
updateTimestamp	String	Update time.
creationTimestamp	String	Creation time.

Table 4-833 PrecheckClusterTask

Parameter	Type	Description
apiVersion	String	API version, which defaults to v3
kind	String	Resource type, which defaults to PreCheckTask
metadata	PrecheckTaskMetadata object	Metadata of a pre-upgrade check task
spec	PrecheckSpec object	Information of a pre-upgrade check task
status	PrecheckStatus object	Status of a pre-upgrade check task

Table 4-834 PrecheckTaskMetadata

Parameter	Type	Description
uid	String	Task ID

Parameter	Type	Description
creationTimes tamp	String	Time when the task was created
updateTimest amp	String	Time when the task was updated

Table 4-835 PrecheckSpec

Parameter	Type	Description
clusterID	String	Cluster ID
clusterVersion	String	Cluster version
targetVersion	String	Target Version
skippedCheckl temList	Array of skippedChec kItem List objects	Skipped check items

Table 4-836 skippedCheckItem

Parameter	Type	Description
name	String	Names of the skipped check items
resourceSelect or	resourceSelec tor object	Resource tag selector. This parameter is available only for node check, but not for cluster check or add-on check.

Table 4-837 resourceSelector

Parameter	Type	Description
key	String	Tag key. Options: <ul style="list-style-type: none"> • node.uid: node UID
values	Array of strings	Tag values
operator	String	Logical operator of a tag. Options: <ul style="list-style-type: none"> • In

Table 4-838 PrecheckStatus

Parameter	Type	Description
phase	String	Status. Options: <ul style="list-style-type: none"> • Init: initializing • Running • Success • Failed • Error
expireTimeStamp	String	Check result expiration time
message	String	Information, which typically indicates the log for execution errors
clusterCheckStatus	clusterCheckStatus object	Status of a cluster restrictions check
addonCheckStatus	addonCheckStatus object	Status of an add-on check
nodeCheckStatus	nodeCheckStatus object	Node check status

Table 4-839 clusterCheckStatus

Parameter	Type	Description
phase	String	Status. Options: <ul style="list-style-type: none"> • Init: initializing • Running • Success • Failed
itemsStatus	Array of PreCheckItemStatus objects	Compliance set by check item

Table 4-840 addonCheckStatus

Parameter	Type	Description
phase	String	Status. Options: <ul style="list-style-type: none"> • Init: initializing • Running • Success • Failed
itemsStatus	Array of PreCheckItemStatus objects	Compliance set by check item

Table 4-841 nodeCheckStatus

Parameter	Type	Description
phase	String	Status. Options: <ul style="list-style-type: none"> • Init: initializing • Running • Success • Failed
nodeStageStatus	Array of NodeStageStatus objects	Node check status

Table 4-842 NodeStageStatus

Parameter	Type	Description
nodeInfo	NodeInfo object	Node information
itemsStatus	Array of PreCheckItemStatus objects	Compliance set by check item

Table 4-843 NodeInfo

Parameter	Type	Description
uid	String	Node UID

Parameter	Type	Description
name	String	Node name
status	String	Status
nodeType	String	Node type

Table 4-844 PreCheckItemStatus

Parameter	Type	Description
name	String	Check item
kind	String	Check item type. Options: <ul style="list-style-type: none"> ● Exception: exceptions that need to be resolved by you ● Risk: risks that can be skipped after you confirm them
group	String	Check item group. Options: <ul style="list-style-type: none"> ● LimitCheck: cluster limit check ● MasterCheck: master node check ● NodeCheck: worker node check ● AddonCheck: add-on check ● ExecuteException: process error check
level	String	Check item severity. Options: <ul style="list-style-type: none"> ● Info: information ● Warning ● Fatal: critical
phase	String	Status. Options: <ul style="list-style-type: none"> ● Init: initializing ● Running ● Success ● Failed
message	String	Information
riskSource	riskSource object	Risk item
errorCodes	Array of strings	Error code set

Table 4-845 riskSource

Parameter	Type	Description
configurationRisks	Array of configurationRisks objects	Configuration risk
deprecatedAPIRisks	Array of deprecatedAPIRisks objects	Deprecated API risk
nodeRisks	Array of nodeRisks objects	Node risk
addonRisks	Array of addonRisks objects	Add-on risk

Table 4-846 configurationRisks

Parameter	Type	Description
package	String	Component
sourceFile	String	How to Obtain
nodeMsg	String	Node information
field	String	Parameter Value
operation	String	Change an operation type.
originalValue	String	Original Value
value	String	Current Value

Table 4-847 deprecatedAPIRisks

Parameter	Type	Description
url	String	Request path, for example, /apis/policy/v1beta1/podsecuritypolicies
userAgent	String	Client information

Table 4-848 nodeRisks

Parameter	Type	Description
NodeID	String	Worker node ID

Table 4-849 addonRisks

Parameter	Type	Description
addonTemplate	String	Name of the add-on template
alias	String	Add-on alias

Example Requests

None

Example Responses

Status code: 200

The list of pre-upgrade check tasks of a cluster is obtained.

```
{
  "kind": "List",
  "apiVersion": "v3",
  "metadata": { },
  "items": [ {
    "kind": "PreCheckTask",
    "apiVersion": "v3",
    "metadata": {
      "uid": "10b52d23-080a-4b7d-bf83-64b4687ca786",
      "creationTimestamp": "2023-12-16 07:07:11.099111 +0000 UTC",
      "updateTimestamp": "2023-12-16 07:09:10.425622 +0000 UTC"
    },
    "spec": {
      "clusterVersion": "v1.23.5-r0",
      "targetVersion": "v1.23.11-r0"
    },
    "status": {
      "phase": "Failed",
      "clusterCheckStatus": {
        "phase": "Success",
        "itemsStatus": [ {
          "name": "DeprecatedApiCheck",
          "kind": "Risk",
          "group": "LimitCheck",
          "level": "Info",
          "phase": "Success",
          "message": "check item succeed",
          "riskSource": { }
        }, {
          "name": "BlackLimit",
          "kind": "Exception",
          "group": "LimitCheck",
          "level": "Fatal",
          "phase": "Success",
          "message": "check item succeed",

```

```

    "riskSource" : { }
  }, {
    "name" : "MasterSSH",
    "kind" : "Exception",
    "group" : "LimitCheck",
    "level" : "Fatal",
    "phase" : "Success",
    "message" : "check item succeed",
    "riskSource" : { }
  }, {
    "name" : "ReleaseLimit",
    "kind" : "Exception",
    "group" : "LimitCheck",
    "level" : "Warning",
    "phase" : "Success",
    "message" : "check item succeed",
    "riskSource" : { }
  }, {
    "name" : "ClusterNoArm",
    "kind" : "Exception",
    "group" : "LimitCheck",
    "level" : "Warning",
    "phase" : "Success",
    "message" : "check item succeed",
    "riskSource" : { }
  }
]
},
"addonCheckStatus" : {
  "phase" : "Failed",
  "itemsStatus" : [ {
    "name" : "AddonLimit",
    "kind" : "Exception",
    "group" : "AddonCheck",
    "level" : "Warning",
    "phase" : "Failed",
    "message" : "addon [ CoreDNS,CCE Container Storage (Everest) ] status is abnormal, check and try
again",
    "riskSource" : {
      "addonRisks" : [ {
        "addonTemplateName" : "coredns",
        "alias" : "CoreDNS"
      }, {
        "addonTemplateName" : "everest",
        "alias" : "CCE Container Storage (Everest)"
      }
    ]
  }
  }, {
    "name" : "CoreDNSConfLimit",
    "kind" : "Exception",
    "group" : "AddonCheck",
    "level" : "Fatal",
    "phase" : "Success",
    "message" : "check item succeed",
    "riskSource" : { }
  }, {
    "name" : "EverestLimitHungVersion",
    "kind" : "Risk",
    "group" : "AddonCheck",
    "level" : "Fatal",
    "phase" : "Success",
    "message" : "check item succeed",
    "riskSource" : { }
  }
]
},
"nodeCheckStatus" : {
  "phase" : "Success"
}
}

```

```
}]
}
```

Status Codes

Status Code	Description
200	The list of pre-upgrade check tasks of a cluster is obtained.

Error Codes

See [Error Codes](#).

4.7.10 Post-upgrade Check

Function

Post-upgrade check, which is used by the customer to check the cluster status after an upgrade and provide feedback after services are restored. Use this API with the console.

Calling Method

For details, see [Calling APIs](#).

URI

POST /api/v3/projects/{project_id}/clusters/{cluster_id}/operation/postcheck

Table 4-850 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-851 Request body parameters

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	API version, which defaults to v3
kind	Yes	String	Resource type
spec	Yes	PostcheckSpec object	spec indicates the configuration confirmed after an upgrade.

Table 4-852 PostcheckSpec

Parameter	Mandatory	Type	Description
clusterID	No	String	Cluster ID
clusterVersion	No	String	Source cluster version
targetVersion	No	String	Target cluster version

Response Parameters

Status code: 200

Table 4-853 Response body parameters

Parameter	Type	Description
apiVersion	String	API version
kind	String	Resource type
metadata	PostcheckClusterResponseMetadata object	Post-upgrade check metadata
spec	PostcheckSpec object	Configuration confirmed after a cluster upgrade
status	status object	Status confirmed after a cluster upgrade

Table 4-854 PostcheckCluserResponseMetadata

Parameter	Type	Description
uid	String	Task ID

Table 4-855 PostcheckSpec

Parameter	Type	Description
clusterID	String	Cluster ID
clusterVersion	String	Source cluster version
targetVersion	String	Target cluster version

Table 4-856 status

Parameter	Type	Description
phase	String	Status. Options: <ul style="list-style-type: none"> • Success • Failed • Error

Example Requests

Post-upgrade check

POST /api/v3/projects/{project_id}/clusters/{cluster_id}/operation/postcheck

```
{
  "kind": "PostCheckTask",
  "apiVersion": "v3",
  "spec": {
    "clusterID": "8978deaa-1743-11ee-8e46-0255ac10004c",
    "clusterVersion": "v1.15.11-r1",
    "targetVersion": "v1.19.16-r80"
  }
}
```

Example Responses

Status code: 200

Cluster post-upgrade check succeeded.

```
{
  "kind": "PostCheckTask",
  "apiVersion": "v3",
  "metadata": {
    "uid": "e99fedf8-348c-4084-b0fd-81bf187df4e0"
  },
  "spec": {
```

```

"clusterID" : "8978deaa-1743-11ee-8e46-0255ac10004c",
"clusterVersion" : "v1.15.11-r1",
"targetVersion" : "v1.19.16-r80"
},
"status" : {
  "phase" : "Success"
}
}

```

Status Codes

Status Code	Description
200	Cluster post-upgrade check succeeded.

Error Codes

See [Error Codes](#).

4.7.11 Cluster Backup

Function

Cluster backup

Calling Method

For details, see [Calling APIs](#).

URI

POST /api/v3.1/projects/{project_id}/clusters/{cluster_id}/operation/snapshot

Table 4-857 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

None

Response Parameters

Status code: 200

Table 4-858 Response body parameters

Parameter	Type	Description
uid	String	Task ID
metadata	SnapshotClusterResponseMetadata object	Backup metadata

Table 4-859 SnapshotClusterResponseMetadata

Parameter	Type	Description
apiVersion	String	API version, which defaults to v3.1
kind	String	Task type

Example Requests

Example of a cluster upgrade backup request

```
POST /api/v3.1/projects/{project_id}/clusters/{cluster_id}/operation/snapshot
```

Example Responses

Status code: 200

The cluster backup task is created.

```
{
  "uid" : "15376f1b-daa6-4e2d-96a6-e9d5d7caeea2",
  "metadata" : {
    "kind" : "Snapshot",
    "apiVersion" : "v3.1"
  }
}
```

Status Codes

Status Code	Description
200	The cluster backup task is created.

Error Codes

See [Error Codes](#).

4.7.12 Obtaining a List of Cluster Backup Task Details

Function

Obtain a list of cluster backup task details.

Calling Method

For details, see [Calling APIs](#).

URI

GET /api/v3.1/projects/{project_id}/clusters/{cluster_id}/operation/snapshot/tasks

Table 4-860 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

None

Response Parameters

Status code: 200

Table 4-861 Response body parameters

Parameter	Type	Description
apiVersion	String	API version, which defaults to v3.1
kind	String	Task type
metadata	SnapshotTaskMetadata object	Backup metadata

Parameter	Type	Description
items	Array of SnapshotTask objects	Backup tasks
status	SnapshotTaskStatus object	Backup task status

Table 4-862 SnapshotTask

Parameter	Type	Description
kind	String	Task type
apiVersion	String	API version
metadata	SnapshotTaskMetadata object	Backup metadata
spec	SnapshotSpec object	Backing Up Task Configurations (To Be Deprecated)
status	SnapshotStatus object	Backup task status

Table 4-863 SnapshotTaskMetadata

Parameter	Type	Description
uid	String	Task ID
creationTimestamp	String	Time when the task was created
updateTimestamp	String	Time when the task was updated

Table 4-864 SnapshotSpec

Parameter	Type	Description
items	Array of SnapshotSpecItems objects	Backup task details

Table 4-865 SnapshotSpecItems

Parameter	Type	Description
id	String	Subtask ID
type	String	Subtask type
status	String	Status
creationTimes tamp	String	Time when the task was created
updateTimest amp	String	Time when the task was updated
message	String	Information

Table 4-866 SnapshotStatus

Parameter	Type	Description
phase	String	Task status
progress	String	Task progress
completionTi me	String	Completion time

Table 4-867 SnapshotTaskStatus

Parameter	Type	Description
latestBackupT ime	String	Time of the latest backup

Example Requests

None

Example Responses

Status code: 200

The list of cluster backup task details is obtained.

```
{
  "kind" : "List",
  "apiVersion" : "v3.1",
  "metadata" : { },
  "items" : [ {
    "kind" : "SnapshotTask",
    "apiVersion" : "v3.1",
```

```
"metadata" : {
  "uid" : "87d326f9-46b0-486e-a4ba-1f82ec9315ed",
  "creationTimestamp" : "2023-11-25 17:03:46.739012 +0800 CST",
  "updateTimestamp" : "2023-11-25 17:03:46.739027 +0800 CST"
},
"spec" : { },
"status" : {
  "phase" : "Running",
  "progress" : "67",
  "completionTime" : "2023-11-25 17:03:46.739027 +0800 CST"
}
}],
"status" : {
  "latestBackupTime" : "2023-11-25 17:03:47.980844 +0800 CST"
}
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ListClusterMasterSnapshotTasksSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        ListClusterMasterSnapshotTasksRequest request = new ListClusterMasterSnapshotTasksRequest();
        request.withClusterId("{cluster_id}");
        try {
            ListClusterMasterSnapshotTasksResponse response = client.listClusterMasterSnapshotTasks(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
        }
    }
}
```

```
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListClusterMasterSnapshotTasksRequest()
        request.cluster_id = "{cluster_id}"
        response = client.list_cluster_master_snapshot_tasks(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
```

```

WithProjectId(projectId).
Build()

client := cce.NewCceClient(
    cce.CceClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListClusterMasterSnapshotTasksRequest{}
request.ClusterId = "{cluster_id}"
response, err := client.ListClusterMasterSnapshotTasks(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The list of cluster backup task details is obtained.

Error Codes

See [Error Codes](#).

4.7.13 Obtaining the Cluster Upgrade Information

Function

Obtain the cluster upgrade information.

Calling Method

For details, see [Calling APIs](#).

URI

GET /api/v3/projects/{project_id}/clusters/{cluster_id}/upgradeinfo

Table 4-868 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

None

Response Parameters

Status code: 200

Table 4-869 Response body parameters

Parameter	Type	Description
kind	String	Type
apiVersion	String	API version
metadata	Metadata object	Basic information about the object. Metadata is a collection of attributes.
spec	UpgradeInfo Spec object	Upgrade settings
status	UpgradeInfo Status object	Upgrade status

Table 4-870 Metadata

Parameter	Type	Description
uid	String	Unique ID.
name	String	Resource name
labels	Map<String,String>	Resource labels in key-value pairs. This is a reserved field and does not take effect.
annotations	Map<String,String>	Resource annotations in the format of key-value pairs.

Parameter	Type	Description
updateTimestamp	String	Update time.
creationTimestamp	String	Creation time.

Table 4-871 UpgradeInfoSpec

Parameter	Type	Description
lastUpgradeInfo	UpgradeInfoStatus object	Last cluster upgrade
versionInfo	UpgradeVersionInfo object	Version
upgradeFeatureGates	UpgradeFeatureGates object	Cluster upgrade feature gates

Table 4-872 UpgradeVersionInfo

Parameter	Type	Description
release	String	Official version, for example, v1.19.10
patch	String	Patch version, for example, r0
suggestPatch	String	Recommended target patch version, for example, r0
targetVersions	Array of strings	Target versions for the upgrade

Table 4-873 UpgradeFeatureGates

Parameter	Type	Description
supportUpgradePageV4	Boolean	Whether the cluster upgrade console supports v4. This field is used by the CCE console.

Table 4-874 UpgradeInfoStatus

Parameter	Type	Description
phase	String	Status of the upgrade task <ul style="list-style-type: none"> • Init: initializing • Running • Pause • Success • Failed
progress	String	Upgrade task progress
completionTime	String	End time of the upgrade task

Example Requests

None

Example Responses

Status code: 200

The cluster upgrade information is obtained.

```
{
  "kind": "UpgradeInfo",
  "apiVersion": "v3",
  "metadata": { },
  "spec": {
    "lastUpgradeInfo": {
      "phase": "Success",
      "completionTime": "2023-11-25 11:18:54.478926 +0800 CST"
    },
    "versionInfo": {
      "release": "v1.27.2",
      "patch": "r0",
      "suggestPatch": "r0",
      "targetVersions": [ "v1.27.3-r0" ]
    },
    "upgradeFeatureGates": {
      "supportUpgradePageV4": true
    }
  },
  "status": {
    "phase": "Success",
    "completionTime": "2023-11-25 11:18:54.478926 +0800 CST"
  }
}
```

Status Codes

Status Code	Description
200	The cluster upgrade information is obtained.

Error Codes

See [Error Codes](#).

4.7.14 Obtaining a Cluster Upgrade Path

Function

Obtain the cluster upgrade path.

Calling Method

For details, see [Calling APIs](#).

URI

GET /api/v3/clusterupgradepaths

Request Parameters

Table 4-875 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-876 Response body parameters

Parameter	Type	Description
apiVersion	String	API version
kind	String	Resource type
metadata	Metadata object	Basic information about the object. Metadata is a collection of attributes.

Parameter	Type	Description
upgradePaths	Array of UpgradePath objects	Upgrade paths

Table 4-877 Metadata

Parameter	Type	Description
uid	String	Unique ID.
name	String	Resource name
labels	Map<String,String>	Resource labels in key-value pairs. This is a reserved field and does not take effect.
annotations	Map<String,String>	Resource annotations in the format of key-value pairs.
updateTimestamp	String	Update time.
creationTimestamp	String	Creation time.

Table 4-878 UpgradePath

Parameter	Type	Description
version	String	Cluster version. For example, v1.19.16-r20 for clusters of v1.19 or earlier, and v1.21 for clusters of v1.21 or later. For details, see CCE Cluster Versions.
platformVersion	String	CCE cluster platform version, which is an internal version under the cluster version (version). Platform versions are used to trace iterations in a major cluster version. They are unique within a major cluster version and recounted when the major cluster version changes. The format of platformVersion is cce.X.Y - X : an internal feature version. When features or OSs in a cluster version are modified or patches are released in the cluster, the value of this parameter increases monotonically from 1.- Y : a patch version of an internal feature version. This parameter is only used for updating a software package for a released feature version. The value of this parameter increases monotonically from 0.

Parameter	Type	Description
targetVersions	Array of strings	Target versions that can be upgraded

Example Requests

None

Example Responses

Status code: 200

The cluster upgrade path is obtained.

```
{
  "kind": "ClusterUpgradePaths",
  "apiVersion": "v3",
  "metadata": { },
  "upgradePaths": [ {
    "version": "v1.25",
    "platformVersion": "cce.5.0",
    "targetVersions": [ "v1.25.6-r0", "v1.27.3-r0" ]
  }, {
    "version": "v1.25",
    "platformVersion": "cce.4.0",
    "targetVersions": [ "v1.25.6-r0", "v1.27.3-r0" ]
  }, {
    "version": "v1.23",
    "platformVersion": "cce.10.0",
    "targetVersions": [ "v1.23.11-r0", "v1.25.6-r0", "v1.27.3-r0" ]
  }, {
    "version": "v1.23",
    "platformVersion": "cce.9.0",
    "targetVersions": [ "v1.23.11-r0", "v1.25.6-r0", "v1.27.3-r0" ]
  } ]
}
```

Status Codes

Status Code	Description
200	The cluster upgrade path is obtained.

Error Codes

See [Error Codes](#).

4.7.15 Obtaining the Configuration of Cluster Upgrade Feature Gates

Function

Obtain the configuration of cluster upgrade feature gates.

Calling Method

For details, see [Calling APIs](#).

URI

GET /api/v3/clusterupgradefeaturegates

Request Parameters

Table 4-879 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-880 Response body parameters

Parameter	Type	Description
apiVersion	String	API version
kind	String	Resource type
metadata	Metadata object	Basic information about the object. Metadata is a collection of attributes.

Parameter	Type	Description
upgradeFeatureGates	Map<String,String>	<p>Feature gates information, which is in the format of a key-value pair.</p> <ul style="list-style-type: none"> • Key: can be DisplayPreCheckDetail (displaying details about all cluster check items before an upgrade), EvsSnapshot (using EVS snapshots to back up clusters), LabelForSkippedNode (labeling the nodes skipped during a cluster upgrade), or UpgradeStrategy (specifying cluster upgrade policies) • Value: can be Support, Disable, or Default, which is determined using default CCE rules.

Table 4-881 Metadata

Parameter	Type	Description
uid	String	Unique ID.
name	String	Resource name
labels	Map<String,String>	Resource labels in key-value pairs. This is a reserved field and does not take effect.
annotations	Map<String,String>	Resource annotations in the format of key-value pairs.
updateTimestamp	String	Update time.
creationTimestamp	String	Creation time.

Example Requests

None

Example Responses

Status code: 200

The cluster upgrade path is obtained.

```
{
  "kind": "ClusterUpgradeFeatureGates",
  "apiVersion": "v3",
  "metadata": { },
  "upgradeFeatureGates": {
    "DisplayPreCheckDetail": "Support",
    "EvsSnapshot": "Support",
    "LabelForSkippedNode": "Support",
```

```
"UpgradeStrategy" : "Support"
}
}
```

Status Codes

Status Code	Description
200	The cluster upgrade path is obtained.

Error Codes

See [Error Codes](#).

4.7.16 Enabling the Cluster Upgrade Process Booting Task

Function

This API is used to create a cluster upgrade booting task. After the booting task is created by calling this API, start the check through the cluster pre-upgrade check.

The upgrade process task is used to control the execution process of the cluster upgrade task. The execution process is as follows: Pre-upgrade check -> Cluster upgrade -> Post-upgrade check.

Calling Method

For details, see [Calling APIs](#).

URI

POST /api/v3/projects/{project_id}/clusters/{cluster_id}/operation/upgradeworkflows

Table 4-882 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-883 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Table 4-884 Request body parameters

Parameter	Mandatory	Type	Description
kind	Yes	String	API type. The value is fixed at WorkflowTask and cannot be changed.
apiVersion	Yes	String	API version. The value is fixed at v3 and cannot be changed.
spec	Yes	WorkflowSpec object	

Table 4-885 WorkflowSpec

Parameter	Mandatory	Type	Description
clusterID	No	String	Cluster ID, which uniquely identifies a resource. The value is automatically generated after the object is created. A custom value will not take effect.
clusterVersion	No	String	Current version of the cluster to be upgraded
targetVersion	Yes	String	Target version of the cluster

Response Parameters

Status code: 201

Table 4-886 Response body parameters

Parameter	Type	Description
kind	String	API type. The value is fixed at WorkflowTask and cannot be changed.
apiVersion	String	API version. The value is fixed at v3 and cannot be changed.
metadata	Metadata object	Upgrade process metadata
spec	WorkflowSpec object	Element type of the collection class. The main body of the cluster upgrade process is provided in spec . CCE creates or updates objects by defining or updating spec .
status	WorkflowStatus object	Element type of the collection class, which is used to record the current status of the cluster upgrade process, including the execution status of each process in the cluster upgrade process.

Table 4-887 Metadata

Parameter	Type	Description
uid	String	Unique ID.
name	String	Resource name
labels	Map<String,String>	Resource labels in key-value pairs. This is a reserved field and does not take effect.
annotations	Map<String,String>	Resource annotations in the format of key-value pairs.
updateTimestamp	String	Update time.
creationTimestamp	String	Creation time.

Table 4-888 WorkflowSpec

Parameter	Type	Description
clusterID	String	Cluster ID, which uniquely identifies a resource. The value is automatically generated after the object is created. A custom value will not take effect.
clusterVersion	String	Current version of the cluster to be upgraded

Parameter	Type	Description
targetVersion	String	Target version of the cluster

Table 4-889 WorkFlowStatus

Parameter	Type	Description
phase	String	Execution status of the cluster upgrade process. Options: Init: No task is running in the upgrade process. Running: Some tasks in the upgrade process are being executed. Pending: A task in the upgrade process fails to be executed. Success: All tasks in the upgrade process have been executed. Cancel: The upgrade process has been canceled.
pointStatuses	Array of PointStatus objects	Execution status of each task in the upgrade process
lineStatuses	Array of LineStatus objects	Task execution path of the upgrade process

Table 4-890 PointStatus

Parameter	Type	Description
taskType	String	Cluster upgrade task type. Options: Cluster: a cluster upgrade task PreCheck: a cluster pre-upgrade check task Rollback: a cluster rollback task Snapshot: a cluster upgrade snapshot task PostCheck: a cluster post-upgrade check task
taskID	String	Upgrade task ID

Parameter	Type	Description
status	String	Cluster upgrade status. Options: Init: The task is in the initial state. Queuing: The task is in the execution queue. Running: The task is being executed. Success: The task has been executed. Failed: Executing the task failed.
startTimeStamp	String	Start time of the upgrade task
endTimeStamp	String	End time of the upgrade task
expireTimeStamp	String	Expiration time of the upgrade task (only for pre-upgrade check tasks)

Table 4-891 LineStatus

Parameter	Type	Description
startPoint	Point object	Path startpoint
endPoint	Point object	Path endpoint
critical	String	Whether it is a key path. An upgrade can be canceled only after the key path is executed.

Table 4-892 Point

Parameter	Type	Description
taskType	String	Cluster upgrade task type

Example Requests

Enable the process of upgrading a cluster to v1.23.

```
POST /api/v3/projects/{project_id}/clusters/{cluster_id}/operation/upgradeworkflows
```

```
{
  "kind": "WorkFlowTask",
  "apiVersion": "v3",
  "spec": {
    "targetVersion": "v1.23"
  }
}
```

Example Responses

Status code: 201

The upgrade process is created in the specified cluster.

```
{
  "kind": "WorkFlowTask",
  "apiVersion": "v3",
  "metadata": {
    "uid": "5ddfddfe-87db-11ec-b5e5-0255ac111914"
  },
  "spec": {
    "clusterID": "b4b9e60f-8aa2-11ee-af09-0255ac10004f",
    "clusterVersion": "v1.17.17-r0",
    "targetVersion": "v1.19.16-r80"
  },
  "status": {
    "pointStatuses": [ {
      "taskType": "PreCheck"
    }, {
      "taskType": "Snapshot"
    }, {
      "taskType": "Cluster"
    }, {
      "taskType": "PostCheck"
    } ],
    "lineStatuses": [ {
      "startPoint": {
        "taskType": "PreCheck"
      },
      "endPoint": {
        "taskType": "Cluster"
      }
    }, {
      "startPoint": {
        "taskType": "Cluster"
      },
      "endPoint": {
        "taskType": "PostCheck"
      }
    } ]
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Enable the process of upgrading a cluster to v1.23.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class CreateUpgradeWorkFlowSolution {
```

```
public static void main(String[] args) {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running
    // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    String ak = System.getenv("CLOUD_SDK_AK");
    String sk = System.getenv("CLOUD_SDK_SK");
    String projectId = "{project_id}";

    ICredential auth = new BasicCredentials()
        .withProjectId(projectId)
        .withAk(ak)
        .withSk(sk);

    CceClient client = CceClient.newBuilder()
        .withCredential(auth)
        .withRegion(CceRegion.valueOf("<YOUR REGION>"))
        .build();
    CreateUpgradeWorkFlowRequest request = new CreateUpgradeWorkFlowRequest();
    request.withClusterId("{cluster_id}");
    CreateUpgradeWorkFlowRequestBody body = new CreateUpgradeWorkFlowRequestBody();
    WorkFlowSpec specbody = new WorkFlowSpec();
    specbody.withTargetVersion("v1.23");
    body.withSpec(specbody);
    body.withApiVersion("v3");
    body.withKind("WorkFlowTask");
    request.withBody(body);
    try {
        CreateUpgradeWorkFlowResponse response = client.createUpgradeWorkFlow(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

Enable the process of upgrading a cluster to v1.23.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \
    client = CceClient.new_builder() \
```

```
.with_credentials(credentials) \  
.with_region(CceRegion.value_of("<YOUR REGION>")) \  
.build()  
  
try:  
    request = CreateUpgradeWorkFlowRequest()  
    request.cluster_id = "{cluster_id}"  
    specbody = WorkFlowSpec(  
        target_version="v1.23"  
    )  
    request.body = CreateUpgradeWorkFlowRequestBody(  
        spec=specbody,  
        api_version="v3",  
        kind="WorkFlowTask"  
    )  
    response = client.create_upgrade_work_flow(request)  
    print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

Go

Enable the process of upgrading a cluster to v1.23.

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()  
  
    client := cce.NewCceClient(  
        cce.CceClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.CreateUpgradeWorkFlowRequest{}  
    request.ClusterId = "{cluster_id}"  
    specbody := &model.WorkFlowSpec{  
        TargetVersion: "v1.23",  
    }  
    request.Body = &model.CreateUpgradeWorkFlowRequestBody{  
        Spec: specbody,  
        ApiVersion: "v3",  
        Kind: "WorkFlowTask",  
    }  
}
```

```

response, err := client.CreateUpgradeWorkFlow(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
201	The upgrade process is created in the specified cluster.

Error Codes

See [Error Codes](#).

4.7.17 Obtaining a List of Upgrade Workflows

Function

Obtain a list of historical cluster upgrade booting tasks.

Calling Method

For details, see [Calling APIs](#).

URI

GET /api/v3/projects/{project_id}/clusters/{cluster_id}/operation/upgradeworkflows

Table 4-893 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-894 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-895 Response body parameters

Parameter	Type	Description
kind	String	API type. The value is fixed at List and cannot be changed.
apiVersion	String	API version. The value is fixed at v3 and cannot be changed.
items	UpgradeWorkFlow object	

Table 4-896 UpgradeWorkFlow

Parameter	Type	Description
kind	String	API type. The value is fixed at WorkFlowTask and cannot be changed.
apiVersion	String	API version. The value is fixed at v3 and cannot be changed.
metadata	Metadata object	Upgrade process metadata
spec	WorkFlowSpec object	Element type of the collection class. The main body of the cluster upgrade process is provided in spec . CCE creates or updates objects by defining or updating spec .

Parameter	Type	Description
status	WorkFlowStatus object	Element type of the collection class, which is used to record the current status of the cluster upgrade process, including the execution status of each process in the cluster upgrade process.

Table 4-897 Metadata

Parameter	Type	Description
uid	String	Unique ID.
name	String	Resource name
labels	Map<String,String>	Resource labels in key-value pairs. This is a reserved field and does not take effect.
annotations	Map<String,String>	Resource annotations in the format of key-value pairs.
updateTimestamp	String	Update time.
creationTimestamp	String	Creation time.

Table 4-898 WorkFlowSpec

Parameter	Type	Description
clusterID	String	Cluster ID, which uniquely identifies a resource. The value is automatically generated after the object is created. A custom value will not take effect.
clusterVersion	String	Current version of the cluster to be upgraded
targetVersion	String	Target version of the cluster

Table 4-899 WorkFlowStatus

Parameter	Type	Description
phase	String	Execution status of the cluster upgrade process. Options: Init: No task is running in the upgrade process. Running: Some tasks in the upgrade process are being executed. Pending: A task in the upgrade process fails to be executed. Success: All tasks in the upgrade process have been executed. Cancel: The upgrade process has been canceled.
pointStatuses	Array of PointStatus objects	Execution status of each task in the upgrade process
lineStatuses	Array of LineStatus objects	Task execution path of the upgrade process

Table 4-900 PointStatus

Parameter	Type	Description
taskType	String	Cluster upgrade task type. Options: Cluster: a cluster upgrade task PreCheck: a cluster pre-upgrade check task Rollback: a cluster rollback task Snapshot: a cluster upgrade snapshot task PostCheck: a cluster post-upgrade check task
taskID	String	Upgrade task ID
status	String	Cluster upgrade status. Options: Init: The task is in the initial state. Queuing: The task is in the execution queue. Running: The task is being executed. Success: The task has been executed. Failed: Executing the task failed.
startTimeStamp	String	Start time of the upgrade task
endTimeStamp	String	End time of the upgrade task

Parameter	Type	Description
expireTimeStamp	String	Expiration time of the upgrade task (only for pre-upgrade check tasks)

Table 4-901 LineStatus

Parameter	Type	Description
startPoint	Point object	Path startpoint
endPoint	Point object	Path endpoint
critical	String	Whether it is a key path. An upgrade can be canceled only after the key path is executed.

Table 4-902 Point

Parameter	Type	Description
taskType	String	Cluster upgrade task type

Example Requests

None

Example Responses

Status code: 200

The list of historical cluster upgrade booting tasks is obtained.

```
{
  "apiVersion": "v3",
  "kind": "List",
  "items": [
    {
      "kind": "WorkflowTask",
      "apiVersion": "v3",
      "metadata": {
        "uid": "730f5577-38ef-448c-b4a7-c6878fbefdda",
        "creationTimestamp": "2023-11-24 08:39:15.894417 +0000 UTC",
        "updateTimestamp": "2023-11-25 02:57:25.718567 +0000 UTC"
      },
      "spec": {
        "clusterID": "b4b9e60f-8aa2-11ee-af09-0255ac10004f",
        "clusterVersion": "v1.17.17-r0",
        "targetVersion": "v1.19.16-r80"
      },
      "status": {
        "phase": "Cancel",
        "pointStatuses": [
          {
            "taskType": "PreCheck"
          },
          {
            "taskType": "Snapshot"
          }
        ]
      }
    }
  ]
}
```

```

    "taskType" : "Cluster"
  }, {
    "taskType" : "PostCheck"
  } ],
  "lineStatuses" : [ {
    "startPoint" : {
      "taskType" : "PreCheck"
    },
    "endPoint" : {
      "taskType" : "Cluster"
    }
  }, {
    "startPoint" : {
      "taskType" : "Cluster"
    },
    "endPoint" : {
      "taskType" : "PostCheck"
    }
  } ]
}
}
}
}

```

SDK Sample Code

The SDK sample code is as follows.

Java

```

package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ListUpgradeWorkFlowsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        ListUpgradeWorkFlowsRequest request = new ListUpgradeWorkFlowsRequest();
        request.withClusterId("{cluster_id}");
        try {
            ListUpgradeWorkFlowsResponse response = client.listUpgradeWorkFlows(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {

```

```
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListUpgradeWorkFlowsRequest()
        request.cluster_id = "{cluster_id}"
        response = client.list_upgrade_work_flows(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
```

```

sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := cce.NewCceClient(
    cce.CceClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListUpgradeWorkFlowsRequest{}
request.ClusterId = "{cluster_id}"
response, err := client.ListUpgradeWorkFlows(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The list of historical cluster upgrade booting tasks is obtained.

Error Codes

See [Error Codes](#).

4.7.18 Obtaining Details About a Specified Cluster Upgrade Task

Function

This API is used to obtain details about an upgrade booting task based on the task ID.

Calling Method

For details, see [Calling APIs](#).

URI

GET /api/v3/projects/{project_id}/clusters/{cluster_id}/operation/upgradeworkflows/{upgrade_workflow_id}

Table 4-903 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
upgrade_workflow_id	Yes	String	Booting process ID of a cluster upgrade task. For details about how to obtain the ID, see Obtaining API URI Parameters .

Request Parameters

Table 4-904 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-905 Response body parameters

Parameter	Type	Description
kind	String	API type. The value is fixed at WorkflowTask and cannot be changed.
apiVersion	String	API version. The value is fixed at v3 and cannot be changed.

Parameter	Type	Description
metadata	Metadata object	Upgrade process metadata
spec	WorkFlowSpec object	Element type of the collection class. The main body of the cluster upgrade process is provided in spec . CCE creates or updates objects by defining or updating spec .
status	WorkFlowStatus object	Element type of the collection class, which is used to record the current status of the cluster upgrade process, including the execution status of each process in the cluster upgrade process.

Table 4-906 Metadata

Parameter	Type	Description
uid	String	Unique ID.
name	String	Resource name
labels	Map<String,String>	Resource labels in key-value pairs. This is a reserved field and does not take effect.
annotations	Map<String,String>	Resource annotations in the format of key-value pairs.
updateTimestamp	String	Update time.
creationTimestamp	String	Creation time.

Table 4-907 WorkFlowSpec

Parameter	Type	Description
clusterID	String	Cluster ID, which uniquely identifies a resource. The value is automatically generated after the object is created. A custom value will not take effect.
clusterVersion	String	Current version of the cluster to be upgraded
targetVersion	String	Target version of the cluster

Table 4-908 WorkFlowStatus

Parameter	Type	Description
phase	String	Execution status of the cluster upgrade process. Options: Init: No task is running in the upgrade process. Running: Some tasks in the upgrade process are being executed. Pending: A task in the upgrade process fails to be executed. Success: All tasks in the upgrade process have been executed. Cancel: The upgrade process has been canceled.
pointStatuses	Array of PointStatus objects	Execution status of each task in the upgrade process
lineStatuses	Array of LineStatus objects	Task execution path of the upgrade process

Table 4-909 PointStatus

Parameter	Type	Description
taskType	String	Cluster upgrade task type. Options: Cluster: a cluster upgrade task PreCheck: a cluster pre-upgrade check task Rollback: a cluster rollback task Snapshot: a cluster upgrade snapshot task PostCheck: a cluster post-upgrade check task
taskID	String	Upgrade task ID
status	String	Cluster upgrade status. Options: Init: The task is in the initial state. Queuing: The task is in the execution queue. Running: The task is being executed. Success: The task has been executed. Failed: Executing the task failed.
startTimeStamp	String	Start time of the upgrade task
endTimeStamp	String	End time of the upgrade task

Parameter	Type	Description
expireTimeStamp	String	Expiration time of the upgrade task (only for pre-upgrade check tasks)

Table 4-910 LineStatus

Parameter	Type	Description
startPoint	Point object	Path startpoint
endPoint	Point object	Path endpoint
critical	String	Whether it is a key path. An upgrade can be canceled only after the key path is executed.

Table 4-911 Point

Parameter	Type	Description
taskType	String	Cluster upgrade task type

Example Requests

None

Example Responses

Status code: 200

Details about the cluster upgrade task are obtained.

```
{
  "kind": "WorkFlowTask",
  "apiVersion": "v3",
  "metadata": {
    "uid": "c271e39e-1a6e-4d3d-8fa8-2a36329c68d1",
    "creationTimestamp": "2023-11-25 06:32:34.923248 +0000 UTC",
    "updateTimestamp": "2023-11-25 07:49:30.281911 +0000 UTC"
  },
  "spec": {
    "clusterID": "b4b9e60f-8aa2-11ee-af09-0255ac10004f",
    "clusterVersion": "v1.17.17-r0",
    "targetVersion": "v1.19.16-r80"
  },
  "status": {
    "phase": "Pending",
    "pointStatuses": [ {
      "taskType": "PreCheck",
      "taskID": "f61e008c-1600-41c0-9bde-121de5a30660",
      "status": "Success",
      "startTimeStamp": "2023-11-25 07:20:04.592972 +0000 UTC",
      "endTimeStamp": "2023-11-25 07:21:05.518966 +0000 UTC",
      "expireTimeStamp": "2023-11-25 08:21:05.518966 +0000 UTC"
    } ], {
  }, {
```

```
"taskType": "Snapshot"
}, {
  "taskType": "Cluster",
  "taskId": "6d799ff6-3afe-4242-80b4-6f0a0fa746cb",
  "status": "Failed",
  "startTimeStamp": "2023-11-25 07:49:30.283459 +0000 UTC",
  "endTimeStamp": "2023-11-25 07:58:35.507243 +0000 UTC"
}, {
  "taskType": "PostCheck"
}],
"lineStatuses": [ {
  "startPoint": {
    "taskType": "PreCheck"
  },
  "endPoint": {
    "taskType": "Cluster"
  }
}, {
  "startPoint": {
    "taskType": "Cluster"
  },
  "endPoint": {
    "taskType": "PostCheck"
  }
}
]
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ShowUpgradeWorkFlowSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowUpgradeWorkFlowRequest request = new ShowUpgradeWorkFlowRequest();
```

```
request.withClusterId("{cluster_id}");
request.withUpgradeWorkflowId("{upgrade_workflow_id}");
try {
    ShowUpgradeWorkFlowResponse response = client.showUpgradeWorkFlow(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowUpgradeWorkFlowRequest()
        request.cluster_id = "{cluster_id}"
        request.upgrade_workflow_id = "{upgrade_workflow_id}"
        response = client.show_upgrade_work_flow(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)
```

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowUpgradeWorkFlowRequest{}
    request.ClusterId = "{cluster_id}"
    request.UpgradeWorkflowId = "{upgrade_workflow_id}"
    response, err := client.ShowUpgradeWorkFlow(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Details about the cluster upgrade task are obtained.

Error Codes

See [Error Codes](#).

4.7.19 Updating the Status of a Specified Cluster Upgrade Booting Task

Function

This API is used to update the status of a specified cluster upgrade booting task. This API applies only to the upgrade cancellation process.

When this API is called, the status of the upgrade process booting task cannot be **running**, **success**, or **cancel**, and the status of the upgrade subtask cannot be **running**, **init**, **pause**, or **queue**.

Calling Method

For details, see [Calling APIs](#).

URI

PATCH /api/v3/projects/{project_id}/clusters/{cluster_id}/operation/upgradeworkflows/{upgrade_workflow_id}

Table 4-912 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
upgrade_workf low_id	Yes	String	Booting process ID of a cluster upgrade task. For details about how to obtain the ID, see Obtaining API URI Parameters .

Request Parameters

Table 4-913 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Table 4-914 Request body parameters

Parameter	Mandatory	Type	Description
status	No	status object	Updated workflow status. (Only Cancel is supported.)

Table 4-915 status

Parameter	Mandatory	Type	Description
phase	No	String	Execution status of the cluster upgrade process. Options: Init: No task is running in the upgrade process. Running: Some tasks in the upgrade process are being executed. Pending: A task in the upgrade process fails to be executed. Success: All tasks in the upgrade process have been executed. Cancel: The upgrade process has been canceled.

Response Parameters

Status code: 200

Table 4-916 Response body parameters

Parameter	Type	Description
kind	String	API type. The value is fixed at WorkFlowTask and cannot be changed.
apiVersion	String	API version. The value is fixed at v3 and cannot be changed.
metadata	Metadata object	Upgrade process metadata
spec	WorkFlowSpec object	Element type of the collection class. The main body of the cluster upgrade process is provided in spec . CCE creates or updates objects by defining or updating spec .

Parameter	Type	Description
status	WorkflowStatus object	Element type of the collection class, which is used to record the current status of the cluster upgrade process, including the execution status of each process in the cluster upgrade process.

Table 4-917 Metadata

Parameter	Type	Description
uid	String	Unique ID.
name	String	Resource name
labels	Map<String,String>	Resource labels in key-value pairs. This is a reserved field and does not take effect.
annotations	Map<String,String>	Resource annotations in the format of key-value pairs.
updateTimestamp	String	Update time.
creationTimestamp	String	Creation time.

Table 4-918 WorkflowSpec

Parameter	Type	Description
clusterID	String	Cluster ID, which uniquely identifies a resource. The value is automatically generated after the object is created. A custom value will not take effect.
clusterVersion	String	Current version of the cluster to be upgraded
targetVersion	String	Target version of the cluster

Table 4-919 WorkFlowStatus

Parameter	Type	Description
phase	String	Execution status of the cluster upgrade process. Options: Init: No task is running in the upgrade process. Running: Some tasks in the upgrade process are being executed. Pending: A task in the upgrade process fails to be executed. Success: All tasks in the upgrade process have been executed. Cancel: The upgrade process has been canceled.
pointStatuses	Array of PointStatus objects	Execution status of each task in the upgrade process
lineStatuses	Array of LineStatus objects	Task execution path of the upgrade process

Table 4-920 PointStatus

Parameter	Type	Description
taskType	String	Cluster upgrade task type. Options: Cluster: a cluster upgrade task PreCheck: a cluster pre-upgrade check task Rollback: a cluster rollback task Snapshot: a cluster upgrade snapshot task PostCheck: a cluster post-upgrade check task
taskID	String	Upgrade task ID
status	String	Cluster upgrade status. Options: Init: The task is in the initial state. Queuing: The task is in the execution queue. Running: The task is being executed. Success: The task has been executed. Failed: Executing the task failed.
startTimeStamp	String	Start time of the upgrade task
endTimeStamp	String	End time of the upgrade task

Parameter	Type	Description
expireTimeStamp	String	Expiration time of the upgrade task (only for pre-upgrade check tasks)

Table 4-921 LineStatus

Parameter	Type	Description
startPoint	Point object	Path startpoint
endPoint	Point object	Path endpoint
critical	String	Whether it is a key path. An upgrade can be canceled only after the key path is executed.

Table 4-922 Point

Parameter	Type	Description
taskType	String	Cluster upgrade task type

Example Requests

Cancel the upgrade process.

```
PATCH /api/v3/projects/47eb1d64cbeb45cfa01ae20af4f4b563/clusters/
f9960c6b-8e60-11ee-9754-0255ac100b05/operation/upgradeworkflows/
d0b7e319-8172-424c-86ea-543cd23f9756
```

```
{
  "status": {
    "phase": "Cancel"
  }
}
```

Example Responses

Status code: 200

The status of the cluster upgrade booting task is updated.

```
{
  "kind": "WorkflowTask",
  "apiVersion": "v3",
  "metadata": {
    "uid": "c271e39e-1a6e-4d3d-8fa8-2a36329c68d1",
    "creationTimestamp": "2023-11-25 06:32:34.923248 +0000 UTC",
    "updateTimestamp": "2023-11-25 07:49:30.281911 +0000 UTC"
  },
  "spec": {
    "clusterID": "b4b9e60f-8aa2-11ee-af09-0255ac10004f",
    "clusterVersion": "v1.17.17-r0",
    "targetVersion": "v1.19.16-r80"
  },
}
```

```
"status" : {
  "phase" : "Cancel",
  "pointStatuses" : [ {
    "taskType" : "PreCheck",
    "taskID" : "f61e008c-1600-41c0-9bde-121de5a30660",
    "status" : "Success",
    "startTimeStamp" : "2023-11-25 07:20:04.592972 +0000 UTC",
    "endTimeStamp" : "2023-11-25 07:21:05.518966 +0000 UTC",
    "expireTimeStamp" : "2023-11-25 08:21:05.518966 +0000 UTC"
  }, {
    "taskType" : "Snapshot"
  }, {
    "taskType" : "Cluster",
    "taskID" : "6d799ff6-3afe-4242-80b4-6f0a0fa746cb",
    "status" : "Failed",
    "startTimeStamp" : "2023-11-25 07:49:30.283459 +0000 UTC",
    "endTimeStamp" : "2023-11-25 07:58:35.507243 +0000 UTC"
  }, {
    "taskType" : "PostCheck"
  } ],
  "lineStatuses" : [ {
    "startPoint" : {
      "taskType" : "PreCheck"
    },
    "endPoint" : {
      "taskType" : "Cluster"
    }
  }, {
    "startPoint" : {
      "taskType" : "Cluster"
    },
    "endPoint" : {
      "taskType" : "PostCheck"
    }
  } ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Cancel the upgrade process.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class UpgradeWorkFlowUpdateSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
```

```
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

CceClient client = CceClient.newBuilder()
    .withCredential(auth)
    .withRegion(CceRegion.valueOf("<YOUR REGION>"))
    .build();
UpgradeWorkFlowUpdateRequest request = new UpgradeWorkFlowUpdateRequest();
request.withClusterId("{cluster_id}");
request.withUpgradeWorkflowId("{upgrade_workflow_id}");
UpgradeWorkFlowUpdateRequestBody body = new UpgradeWorkFlowUpdateRequestBody();
UpgradeWorkFlowUpdateRequestBodyStatus statusbody = new
UpgradeWorkFlowUpdateRequestBodyStatus();
statusbody.withPhase(UpgradeWorkFlowUpdateRequestBodyStatus.PhaseEnum.fromValue("Cancel"));
body.withStatus(statusbody);
request.withBody(body);
try {
    UpgradeWorkFlowUpdateResponse response = client.upgradeWorkFlowUpdate(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

Cancel the upgrade process.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpgradeWorkFlowUpdateRequest()
        request.cluster_id = "{cluster_id}"
```

```
request.upgrade_workflow_id = "{upgrade_workflow_id}"
statusbody = UpgradeWorkFlowUpdateRequestBodyStatus(
    phase="Cancel"
)
request.body = UpgradeWorkFlowUpdateRequestBody(
    status=statusbody
)
response = client.upgrade_work_flow_update(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

Cancel the upgrade process.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpgradeWorkFlowUpdateRequest{}
    request.ClusterId = "{cluster_id}"
    request.UpgradeWorkflowId = "{upgrade_workflow_id}"
    phaseStatus:= model.GetUpgradeWorkFlowUpdateRequestBodyStatusPhaseEnum().CANCEL
    statusbody := &model.UpgradeWorkFlowUpdateRequestBodyStatus{
        Phase: &phaseStatus,
    }
    request.Body = &model.UpgradeWorkFlowUpdateRequestBody{
        Status: statusbody,
    }
    response, err := client.UpgradeWorkFlowUpdate(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The status of the cluster upgrade booting task is updated.

Error Codes

See [Error Codes](#).

4.8 Quota Management

4.8.1 Querying Resource Quotas

Function

This API is used to query CCE resource quotas.

Calling Method

For details, see [Calling APIs](#).

URI

GET /api/v3/projects/{project_id}/quotas

Table 4-923 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-924 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-925 Response body parameters

Parameter	Type	Description
quotas	Array of QuotaResource objects	Resources

Table 4-926 QuotaResource

Parameter	Type	Description
quotaKey	String	Resource type
quotaLimit	Integer	Quota
used	Integer	Number of created resources
unit	String	Unit
regionId	String	Region ID. This parameter is not returned if not involved.
availabilityZoneId	String	AZ ID. This parameter is not returned if not involved.

Example Requests

None

Example Responses

Status code: 200

The resource quotas are obtained successfully.

```
{
  "quotas" : [ {
    "quotaKey" : "cluster",
    "quotaLimit" : 20,
    "used" : 13,
    "unit" : "count"
  } ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ShowQuotasSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowQuotasRequest request = new ShowQuotasRequest();
        try {
            ShowQuotasResponse response = client.showQuotas(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
        }
    }
}
```

```
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowQuotasRequest()
        response = client.show_quotas(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()
```



```
client := cce.NewCceClient(  
    cce.CceClientBuilder().  
        WithRegion(region.ValueOf("<YOUR REGION>")).  
        WithCredential(auth).  
        Build())  
  
request := &model.ShowQuotasRequest{}  
response, err := client.ShowQuotas(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The resource quotas are obtained successfully.

Error Codes

See [Error Codes](#).

4.9 API Versions

4.9.1 Obtaining API Versions

Function

This API is used to query the list of API versions supported by CCE.

Calling Method

For details, see [Calling APIs](#).

URI

GET /

Request Parameters

None

Response Parameters

Status code: 200

Table 4-927 Response body parameters

Parameter	Type	Description
versions	Array of APIVersionDetail objects	API version list.

Table 4-928 APIVersionDetail

Parameter	Type	Description
id	String	API version ID. Example: v3
links	Array of APIVersionLink objects	URL of the API version.
min_version	String	Microversion. If the APIs of this version support microversions, set this parameter to the supported minimum microversion. If no microversion is supported, leave this parameter blank.
status	String	API version status. Possible values: <ul style="list-style-type: none"> ● CURRENT: preferred API version ● SUPPORTED: old API version that is still supported ● DEPRECATED: discarded API version that will be deleted
updated	String	Time when the API was released (UTC). For example, if the API version is v3, the value is '2018-09-15 00:00:00Z'.
version	String	Microversion. If the APIs of this version support microversions, set this parameter to the supported maximum microversion. If no microversion is supported, leave this parameter blank.

Table 4-929 APIVersionLink

Parameter	Type	Description
href	String	URL of the API version.
rel	String	Link attributes. self : A self link contains a versioned link to the resource. Use these links immediately after linking.

Example Requests

None

Example Responses

Status code: 200

The API version list is queried successfully.

```
{
  "versions": [ {
    "id": "v3",
    "links": [ {
      "href": "https://cce.region.***.com/v3",
      "rel": "self"
    } ],
    "min_version": "",
    "status": "CURRENT",
    "updated": "2018-09-15 00:00:00Z",
    "version": ""
  } ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ShowVersionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
```

```
String sk = System.getenv("CLOUD_SDK_SK");

ICredential auth = new BasicCredentials()
    .withAk(ak)
    .withSk(sk);

CceClient client = CceClient.newBuilder()
    .withCredential(auth)
    .withRegion(CceRegion.valueOf("<YOUR REGION>"))
    .build();
ShowVersionRequest request = new ShowVersionRequest();
try {
    ShowVersionResponse response = client.showVersion(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowVersionRequest()
        response = client.show_version(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
```

```

"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowVersionRequest{}
    response, err := client.ShowVersion(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The API version list is queried successfully.

Error Codes

See [Error Codes](#).

4.10 Tag Management

4.10.1 Adding Resource Tags to a Specified Cluster in Batches

Function

This API is used to add resource tags for a specified cluster in batches.

 NOTE

- Each cluster supports a maximum of 20 resource tags.
- This API is idempotent. If the to-be-added tag has the same tag key and tag value as an existing tag, the tag will be added. If the to-be-added tag has the same key but different value as an existing tag, the tag will overwrite the existing one.

Calling Method

For details, see [Calling APIs](#).

URI

POST /api/v3/projects/{project_id}/clusters/{cluster_id}/tags/create

Table 4-930 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-931 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Table 4-932 Request body parameters

Parameter	Mandatory	Type	Description
tags	Yes	Array of ResourceTag objects	List of cluster resource tags to be created. Each cluster supports a maximum of 20 resource tags.

Table 4-933 ResourceTag

Parameter	Mandatory	Type	Description
key	No	String	Key. <ul style="list-style-type: none"> Cannot be null. Max characters: 128. Use letters, digits, and spaces in UTF-8 format. Can contain the following special characters: <code>._:/=+-@</code>. Cannot start with <code>_sys_</code>.
value	No	String	Value. <ul style="list-style-type: none"> Can be null but not the default. Max characters: 255. Use letters, digits, and spaces in UTF-8 format. Can contain the following special characters: <code>._:/=+-@</code>.

Response Parameters

None

Example Requests

Adding Resource Tags to a Specified Cluster in Batches

```
POST /api/v3/projects/{project_id}/clusters/{cluster_id}/tags/create
```

```
{
  "tags" : [ {
    "key" : "key1",
    "value" : "value1"
  }, {
    "key" : "key2",
    "value" : "value3"
  } ]
}
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

Adding Resource Tags to a Specified Cluster in Batches

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchCreateClusterTagsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchCreateClusterTagsRequest request = new BatchCreateClusterTagsRequest();
        request.withClusterId("{cluster_id}");
        BatchCreateClusterTagsRequestBody body = new BatchCreateClusterTagsRequestBody();
        List<ResourceTag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new ResourceTag()
                .withKey("key1")
                .withValue("value1")
        );
        listbodyTags.add(
            new ResourceTag()
                .withKey("key2")
                .withValue("value3")
        );
        body.withTags(listbodyTags);
        request.withBody(body);
        try {
            BatchCreateClusterTagsResponse response = client.batchCreateClusterTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
```



```
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

Adding Resource Tags to a Specified Cluster in Batches

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchCreateClusterTagsRequest()
        request.cluster_id = "{cluster_id}"
        listTagsbody = [
            ResourceTag(
                key="key1",
                value="value1"
            ),
            ResourceTag(
                key="key2",
                value="value3"
            )
        ]
        request.body = BatchCreateClusterTagsRequestBody(
            tags=listTagsbody
        )
        response = client.batch_create_cluster_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Adding Resource Tags to a Specified Cluster in Batches

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchCreateClusterTagsRequest{}
    request.ClusterId = "{cluster_id}"
    keyTags := "key1"
    valueTags := "value1"
    keyTags1 := "key2"
    valueTags1 := "value3"
    var listTagsbody = []model.ResourceTag{
        {
            Key: &keyTags,
            Value: &valueTags,
        },
        {
            Key: &keyTags1,
            Value: &valueTags1,
        },
    }
    request.Body = &model.BatchCreateClusterTagsRequestBody{
        Tags: listTagsbody,
    }
    response, err := client.BatchCreateClusterTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	No Content

Error Codes

See [Error Codes](#).

4.10.2 Deleting Resource Tags of a Specified Cluster in Batches

Function

This API is used to delete resource tags of a specified cluster in batches.

NOTE

- This API is idempotent. If the tag key of the to-be-deleted tag does not exist, the tag will be deleted.

Calling Method

For details, see [Calling APIs](#).

URI

POST /api/v3/projects/{project_id}/clusters/{cluster_id}/tags/delete

Table 4-934 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-935 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Table 4-936 Request body parameters

Parameter	Mandatory	Type	Description
tags	Yes	Array of ResourceDeleteTag objects	List of cluster resource tags to be deleted

Table 4-937 ResourceDeleteTag

Parameter	Mandatory	Type	Description
key	No	String	<p>The key value of the resource tag</p> <ul style="list-style-type: none"> It cannot be null. It can contain a maximum of 128 characters. It can contain Chinese characters, letters, digits, and spaces in UTF-8 format. It can contain the following special characters: <code>._:/=+-@</code>. It cannot start with <code>_sys_</code>.

Response Parameters

None

Example Requests

Deleting Resource Tags of a Specified Cluster in Batches

```
POST /api/v3/projects/{project_id}/clusters/{cluster_id}/tags/delete
{
  "tags" : [ {
    "key" : "key1"
  }, {
    "key" : "key2"
  } ]
}
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

Deleting Resource Tags of a Specified Cluster in Batches

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchDeleteClusterTagsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchDeleteClusterTagsRequest request = new BatchDeleteClusterTagsRequest();
        request.withClusterId("{cluster_id}");
        BatchDeleteClusterTagsRequestBody body = new BatchDeleteClusterTagsRequestBody();
        List<ResourceDeleteTag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
```

```
        new ResourceDeleteTag()
            .withKey("key1")
    );
    listbodyTags.add(
        new ResourceDeleteTag()
            .withKey("key2")
    );
    body.withTags(listbodyTags);
    request.withBody(body);
    try {
        BatchDeleteClusterTagsResponse response = client.batchDeleteClusterTags(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
    }
}
```

Python

Deleting Resource Tags of a Specified Cluster in Batches

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchDeleteClusterTagsRequest()
        request.cluster_id = "{cluster_id}"
        listTagsbody = [
            ResourceDeleteTag(
                key="key1"
            ),
            ResourceDeleteTag(
                key="key2"
            )
        ]
        request.body = BatchDeleteClusterTagsRequestBody(
            tags=listTagsbody
        )
        response = client.batch_delete_cluster_tags(request)
```

```
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

Deleting Resource Tags of a Specified Cluster in Batches

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchDeleteClusterTagsRequest{}
    request.ClusterId = "{cluster_id}"
    keyTags := "key1"
    keyTags1 := "key2"
    var listTagsbody = []model.ResourceDeleteTag{
        {
            Key: &keyTags,
        },
        {
            Key: &keyTags1,
        },
    }
    request.Body = &model.BatchDeleteClusterTagsRequestBody{
        Tags: listTagsbody,
    }
    response, err := client.BatchDeleteClusterTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	No Content

Error Codes

See [Error Codes](#).

4.11 Configuration Management

4.11.1 Obtaining the Parameters That Can Be Configured for a Node Pool

Function

This API is used to obtain the parameters that can be configured in a specified CCE node pool.

Calling Method

For details, see [Calling APIs](#).

URI

GET /api/v3/projects/{project_id}/clusters/{cluster_id}/nodepools/{nodepool_id}/configuration/detail

Table 4-938 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Parameter	Mandatory	Type	Description
nodepool_id	Yes	String	Node pool ID.

Request Parameters

Table 4-939 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-940 Response body parameters

Parameter	Type	Description
{User defined key}	Map<String,Array< Package Options >>	Response body for obtaining the parameter configuration list of a specified node pool

Table 4-941 PackageOptions

Parameter	Type	Description
name	String	Parameter name
default	Object	Default parameter value. If this parameter is not specified, the default value will take effect. The returned parameter can be of integer, string, or Boolean type.

Parameter	Type	Description
validAt	String	How the parameter takes effect. Options: <ul style="list-style-type: none"> • static: The parameter takes effect when a node is created and cannot be modified later. • immediately: The parameter can be modified when a node is running and takes effect immediately after modification.
empty	Boolean	Whether the configuration item can be empty. Options: <ul style="list-style-type: none"> • true: If this parameter is set to true, no value will take effect. • false: If this parameter is set to false, the default value will take effect.
schema	String	Parameter category
type	String	Parameter type

Example Requests

None

Example Responses

Status code: 200

The parameters of the specified node pool are obtained.

```
{
  "kubelet" : [ {
    "name" : "kube-api-qps",
    "default" : 300,
    "validAt" : "immediately",
    "empty" : false,
    "schema" : "",
    "type" : "float"
  } ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
```

```
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ShowNodePoolConfigurationDetailsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowNodePoolConfigurationDetailsRequest request = new
        ShowNodePoolConfigurationDetailsRequest();
        request.withClusterId("{cluster_id}");
        request.withNodepoolId("{nodepool_id}");
        try {
            ShowNodePoolConfigurationDetailsResponse response =
            client.showNodePoolConfigurationDetails(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \
    client = CceClient.new_builder() \
```

```
.with_credentials(credentials) \
.with_region(CceRegion.value_of("<YOUR REGION>")) \
.build()

try:
    request = ShowNodePoolConfigurationDetailsRequest()
    request.cluster_id = "{cluster_id}"
    request.nodepool_id = "{nodepool_id}"
    response = client.show_node_pool_configuration_details(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowNodePoolConfigurationDetailsRequest{}
    request.ClusterId = "{cluster_id}"
    request.NodepoolId = "{nodepool_id}"
    response, err := client.ShowNodePoolConfigurationDetails(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The parameters of the specified node pool are obtained.

Error Codes

See [Error Codes](#).

4.11.2 Obtaining the List of Parameters That Can Be Configured for a Cluster

Function

This API is used to obtain the list of parameters that can be configured for a CCE cluster.

Calling Method

For details, see [Calling APIs](#).

URI

GET /api/v3/projects/{project_id}/clusters/{cluster_id}/configuration/detail

Table 4-942 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-943 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-944 Response body parameters

Parameter	Type	Description
{User defined key}	Map<String,Array< Package Options >>	Response body for obtaining the parameter configuration list of a cluster

Table 4-945 PackageOptions

Parameter	Type	Description
name	String	Parameter name
default	Object	Default parameter value. If this parameter is not specified, the default value will take effect. The returned parameter can be of integer, string, or Boolean type.
validAt	String	How the parameter takes effect. Options: <ul style="list-style-type: none"> • static: The parameter takes effect when a node is created and cannot be modified later. • immediately: The parameter can be modified when a node is running and takes effect immediately after modification.
empty	Boolean	Whether the configuration item can be empty. Options: <ul style="list-style-type: none"> • true: If this parameter is set to true, no value will take effect. • false: If this parameter is set to false, the default value will take effect.

Parameter	Type	Description
schema	String	Parameter category
type	String	Parameter type

Example Requests

None

Example Responses

Status code: 200

A list of parameters that can be configured for a cluster is obtained successfully.

```
{
  "kube-apiserver" : [ {
    "name" : "default-not-ready-toleration-seconds",
    "default" : 300,
    "validAt" : "immediately",
    "empty" : true,
    "schema" : "kubernetes",
    "type" : "int"
  } ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ShowClusterConfigurationDetailsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);
```

```
CceClient client = CceClient.newBuilder()
    .withCredential(auth)
    .withRegion(CceRegion.valueOf("<YOUR REGION>"))
    .build();
ShowClusterConfigurationDetailsRequest request = new ShowClusterConfigurationDetailsRequest();
request.withClusterId("{cluster_id}");
try {
    ShowClusterConfigurationDetailsResponse response =
client.showClusterConfigurationDetails(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowClusterConfigurationDetailsRequest()
        request.cluster_id = "{cluster_id}"
        response = client.show_cluster_configuration_details(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
```



```

"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowClusterConfigurationDetailsRequest{}
    request.ClusterId = "{cluster_id}"
    response, err := client.ShowClusterConfigurationDetails(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	A list of parameters that can be configured for a cluster is obtained successfully.

Error Codes

See [Error Codes](#).

4.11.3 Obtaining the Parameters That Can Be Configured for a Node Pool

Function

This API is used to obtain the parameters that can be configured for a node pool.

Calling Method

For details, see [Calling APIs](#).

URI

GET /api/v3/projects/{project_id}/clusters/{cluster_id}/nodepools/{nodepool_id}/configuration

Table 4-946 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
nodepool_id	Yes	String	Node pool ID.

Request Parameters

Table 4-947 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-948 Response body parameters

Parameter	Type	Description
apiVersion	String	API version
kind	String	API type. The value is fixed at Configuration .
metadata	ConfigurationMetadata object	Configuration metadata
spec	ClusterConfigurationsSpec object	Configuration specifications
status	Object	Configuration status

Table 4-949 ConfigurationMetadata

Parameter	Type	Description
name	String	Configuration name
labels	Map<String,String>	<p>A configuration label in a key-value pair.</p> <ul style="list-style-type: none"> • Key: Enter 1 to 63 characters, starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain can be prefixed to a key and contain a maximum of 253 characters. Example DNS subdomain: example.com/my-key • Value: The value can be left blank or contain 1 to 63 characters that start with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed in the character string. <p>Example: "foo": "bar"</p>

Table 4-950 ClusterConfigurationsSpec

Parameter	Type	Description
packages	Array of packages objects	Component configuration item details

Table 4-951 packages

Parameter	Type	Description
name	String	Component name
configurations	Array of Configuratio nItem objects	Component configuration items

Table 4-952 ConfigurationItem

Parameter	Type	Description
name	String	Component configuration item name.
value	Object	Component configuration item value.

Example Requests

None

Example Responses

Status code: 200

The parameter configurations of the node pool are obtained successfully.

```
{
  "kind": "Configuration",
  "apiVersion": "v3",
  "metadata": {
    "name": "configuration",
    "labels": {
      "nodepool_id": "61de338d-a1f9-11ed-8891-0255ac100036"
    }
  },
  "spec": {
    "packages": [ {
      "name": "kube-apiserver",
      "configurations": [ {
        "name": "event-rate-limit-qps",
        "value": 200
      }, {
        "name": "support-overload",
        "value": false
      } ]
    }, {
      "name": "kube-scheduler",
      "configurations": [ {
        "name": "kube-api-qps",
        "value": 100
      }, {
        "name": "default-scheduler",
        "value": "kube-scheduler"
      } ]
    } ]
  }
}
```

```
"status" : { }  
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cce.v3.region.CceRegion;  
import com.huaweicloud.sdk.cce.v3.*;  
import com.huaweicloud.sdk.cce.v3.model.*;  
  
public class ShowNodePoolConfigurationsSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        CceClient client = CceClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ShowNodePoolConfigurationsRequest request = new ShowNodePoolConfigurationsRequest();  
        request.withClusterId("{cluster_id}");  
        request.withNodepoolId("{nodepool_id}");  
        try {  
            ShowNodePoolConfigurationsResponse response = client.showNodePoolConfigurations(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

Python

```
# coding: utf-8  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials
```

```
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowNodePoolConfigurationsRequest()
        request.cluster_id = "{cluster_id}"
        request.nodepool_id = "{nodepool_id}"
        response = client.show_node_pool_configurations(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowNodePoolConfigurationsRequest{}
    request.ClusterId = "{cluster_id}"
```

```
request.NodepoolId = "{nodepool_id}"
response, err := client.ShowNodePoolConfigurations(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The parameter configurations of the node pool are obtained successfully.

Error Codes

See [Error Codes](#).

4.11.4 Changing the Values of Configuration Parameters of a Node Pool

Function

This API is used to change the values of parameter configurations of a node pool in the CCE cluster.

Calling Method

For details, see [Calling APIs](#).

URI

PUT /api/v3/projects/{project_id}/clusters/{cluster_id}/nodepools/{nodepool_id}/configuration

Table 4-953 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Parameter	Mandatory	Type	Description
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .
nodepool_id	Yes	String	Node pool ID.

Request Parameters

Table 4-954 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Table 4-955 Request body parameters

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	API version. The value is fixed at v3 .
kind	Yes	String	API type. The value is fixed at Configuration .
metadata	Yes	ConfigurationMetadata object	Configuration metadata
spec	Yes	ClusterConfigurationsSpec object	Configuration specifications

Table 4-956 ConfigurationMetadata

Parameter	Mandatory	Type	Description
name	Yes	String	Configuration name

Parameter	Mandatory	Type	Description
labels	No	Map<String,String>	<p>A configuration label in a key-value pair.</p> <ul style="list-style-type: none"> • Key: Enter 1 to 63 characters, starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain can be prefixed to a key and contain a maximum of 253 characters. Example DNS subdomain: example.com/my-key • Value: The value can be left blank or contain 1 to 63 characters that start with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed in the character string. <p>Example: "foo": "bar"</p>

Table 4-957 ClusterConfigurationsSpec

Parameter	Mandatory	Type	Description
packages	Yes	Array of packages objects	Component configuration item details

Table 4-958 packages

Parameter	Mandatory	Type	Description
name	No	String	Component name
configurations	No	Array of ConfigurationItem objects	Component configuration items

Table 4-959 ConfigurationItem

Parameter	Mandatory	Type	Description
name	No	String	Component configuration item name.
value	No	Object	Component configuration item value.

Response Parameters

Status code: 200

Table 4-960 Response body parameters

Parameter	Type	Description
apiVersion	String	API version
kind	String	API type. The value is fixed at Configuration .
metadata	ConfigurationMetadata object	Configuration metadata
spec	ClusterConfigurationsSpec object	Configuration specifications
status	Object	Configuration status

Table 4-961 ConfigurationMetadata

Parameter	Type	Description
name	String	Configuration name

Parameter	Type	Description
labels	Map<String,String>	<p>A configuration label in a key-value pair.</p> <ul style="list-style-type: none"> • Key: Enter 1 to 63 characters, starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain can be prefixed to a key and contain a maximum of 253 characters. Example DNS subdomain: example.com/my-key • Value: The value can be left blank or contain 1 to 63 characters that start with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed in the character string. <p>Example: "foo": "bar"</p>

Table 4-962 ClusterConfigurationsSpec

Parameter	Type	Description
packages	Array of packages objects	Component configuration item details

Table 4-963 packages

Parameter	Type	Description
name	String	Component name
configurations	Array of ConfigurationItem objects	Component configuration items

Table 4-964 ConfigurationItem

Parameter	Type	Description
name	String	Component configuration item name.
value	Object	Component configuration item value.

Example Requests

The following example describes how to update the **system-reserved-mem** and **kube-reserved-mem** parameters in kubelet.

```
/api/v3/projects/{project_id}/clusters/{cluster_id}/nodepools/{nodepool_id}/configuration

{
  "kind": "Configuration",
  "apiVersion": "v3",
  "metadata": {
    "name": "configuration"
  },
  "spec": {
    "packages": [ {
      "name": "kubelet",
      "configurations": [ {
        "name": "system-reserved-mem",
        "value": 600
      }, {
        "name": "kube-reserved-mem",
        "value": 800
      } ]
    } ]
  }
}
```

Example Responses

Status code: 200

The parameter configurations of the node pool are updated successfully.

```
{
  "kind": "Configuration",
  "apiVersion": "v3",
  "metadata": {
    "name": "configuration",
    "labels": {
      "nodepool_id": "61de338d-a1f9-11ed-8891-0255ac100036"
    }
  },
  "spec": {
    "packages": [ {
      "name": "kube-apiserver",
      "configurations": [ {
        "name": "event-rate-limit-qps",
        "value": 200
      }, {
        "name": "support-overload",
        "value": false
      } ]
    }, {
      "name": "kube-scheduler",
      "configurations": [ {
        "name": "kube-api-qps",
        "value": 100
      }, {
        "name": "default-scheduler",
        "value": "kube-scheduler"
      } ]
    } ]
  },
  "status": { }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

The following example describes how to update the **system-reserved-mem** and **kube-reserved-mem** parameters in kubelet.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateNodePoolConfigurationSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateNodePoolConfigurationRequest request = new UpdateNodePoolConfigurationRequest();
        request.withClusterId("{cluster_id}");
        request.withNodepoolId("{nodepool_id}");
        UpdateClusterConfigurationsBody body = new UpdateClusterConfigurationsBody();
        List listPackagesConfigurations = new ArrayList<>();
        listPackagesConfigurations.add(
            new ConfigurationItem()
                .withName("system-reserved-mem")
                .withValue("600")
        );
        listPackagesConfigurations.add(
            new ConfigurationItem()
                .withName("kube-reserved-mem")
                .withValue("800")
        );
        List listSpecPackages = new ArrayList<>();
        listSpecPackages.add(
            new ClusterConfigurationsSpecPackages()
                .withName("kubelet")
                .withConfigurations(listPackagesConfigurations)
        );
        ClusterConfigurationsSpec specbody = new ClusterConfigurationsSpec();
        specbody.withPackages(listSpecPackages);
        ConfigurationMetadata metadatabody = new ConfigurationMetadata();
```

```
metadatabody.withName("configuration");
body.withSpec(specbody);
body.withMetadata(metadatabody);
body.withKind("Configuration");
body.withApiVersion("v3");
request.withBody(body);
try {
    UpdateNodePoolConfigurationResponse response = client.updateNodePoolConfiguration(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

The following example describes how to update the **system-reserved-mem** and **kube-reserved-mem** parameters in kubelet.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateNodePoolConfigurationRequest()
        request.cluster_id = "{cluster_id}"
        request.nodepool_id = "{nodepool_id}"
        listConfigurationsPackages = [
            ConfigurationItem(
                name="system-reserved-mem",
                value="600"
            ),
            ConfigurationItem(
                name="kube-reserved-mem",
                value="800"
            )
        ]
        listPackagesSpec = [
            ClusterConfigurationsSpecPackages(
                name="kubelet",
```

```
        configurations=listConfigurationsPackages
    )
]
specbody = ClusterConfigurationsSpec(
    packages=listPackagesSpec
)
metadatabody = ConfigurationMetadata(
    name="configuration"
)
request.body = UpdateClusterConfigurationsBody(
    spec=specbody,
    metadata=metadatabody,
    kind="Configuration",
    api_version="v3"
)
response = client.update_node_pool_configuration(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

The following example describes how to update the **system-reserved-mem** and **kube-reserved-mem** parameters in kubelet.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateNodePoolConfigurationRequest{
        request.ClusterId = "{cluster_id}"
        request.NodepoolId = "{nodepool_id}"
        nameConfigurations:= "system-reserved-mem"
        valueConfigurations:= "600"
        var valueConfigurationsInterface interface{} = valueConfigurations
        nameConfigurations1:= "kube-reserved-mem"
        valueConfigurations1:= "800"
        var valueConfigurationsInterface1 interface{} = valueConfigurations1
```

```

var listConfigurationsPackages = []model.ConfigurationItem{
    {
        Name: &nameConfigurations,
        Value: &valueConfigurationsInterface,
    },
    {
        Name: &nameConfigurations1,
        Value: &valueConfigurationsInterface1,
    },
}
namePackages:= "kubelet"
var listPackagesSpec = []model.ClusterConfigurationsSpecPackages{
    {
        Name: &namePackages,
        Configurations: &listConfigurationsPackages,
    },
}
specbody := &model.ClusterConfigurationsSpec{
    Packages: listPackagesSpec,
}
metadatabody := &model.ConfigurationMetadata{
    Name: "configuration",
}
request.Body = &model.UpdateClusterConfigurationsBody{
    Spec: specbody,
    Metadata: metadatabody,
    Kind: "Configuration",
    ApiVersion: "v3",
}
response, err := client.UpdateNodePoolConfiguration(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The parameter configurations of the node pool are updated successfully.

Error Codes

See [Error Codes](#).

4.12 Chart Management

4.12.1 Uploading a Chart

Function

This API is used to upload a chart.

Calling Method

For details, see [Calling APIs](#).

URI

POST /v2/charts

Request Parameters

Table 4-965 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Table 4-966 FormData parameters

Parameter	Mandatory	Type	Description
parameters	No	String	Parameter configurations for uploading a chart. The following is an example: <pre>"{"override":true,"skip_lint":true,"source":"package"}"</pre> <ul style="list-style-type: none"> skip_lint: whether lint uploaded chart - override: whether override existed chart - visible: update chart visible
content	Yes	File	Chart package file

Response Parameters

Status code: 201

Table 4-967 Response body parameters

Parameter	Type	Description
id	String	Chart ID
name	String	Chart name
values	String	Chart value
translate	String	Chart translation resources
instruction	String	Chart description
version	String	Chart version
description	String	Chart description
source	String	Chart source
icon_url	String	URL to chart icons
public	Boolean	Whether the chart is public
chart_url	String	URL to the chart
create_at	String	Created at
update_at	String	Updated at

Example Requests

```
POST /v2/charts
{
  "parameters": "{ \"override\":true, \"skip_lint\":true, \"source\": \"package\" }",
  "content": "chart-file.tgz"
}
```

Example Responses

Status code: 201

Created

```
{
  "id": "e99a7e86-afdd-11eb-aca3-0255ac100b0e",
  "name": "neo4j",
  "values": "{ \"acceptLicenseAgreement\": \"no\", \"affinity\": {}, \"authEnabled\": true, \"clusterDomain\": \"cluster.local\", \"core\": { \"initContainers\": [], \"numberOfServers\": 3, \"persistentVolume\": { \"enabled\": true, \"mountPath\": \"/data\", \"size\": \"10Gi\" }, \"sidecarContainers\": [] }, \"defaultDatabase\": \"neo4j\", \"image\": \"neo4j\", \"imagePullPolicy\": \"IfNotPresent\", \"imageTag\": \"4.0.3-enterprise\", \"name\": \"neo4j\", \"nodeSelector\": {}, \"podDisruptionBudget\": {}, \"readReplica\": { \"autoscaling\": { \"enabled\": false, \"maxReplicas\": 3, \"minReplicas\": 1, \"targetAverageUtilization\": 70 }, \"initContainers\": [], \"numberOfServers\": 0 }, \"resources\": {}, \"sidecarContainers\": [], \"resources\": {}, \"testImage\": \"markhneedham/k8s-kubectl\", \"testImageTag\": \"master\", \"tolerations\": [], \"useAPOC\": true }",
  "translate": ""
}
```

```
"instruction" : "README.md",
"version" : "3.0.1",
"description" : "DEPRECATED Neo4j is the world's leading graph database",
"source" : "",
"icon_url" : "https://info.neo4j.com/rs/773-GON-065/images/neo4j_logo.png",
"public" : false,
"chart_url" : "neo4j-3.0.1.tgz",
"create_at" : "2021-05-08T08:53:13Z",
"update_at" : "2021-05-08T08:53:13Z"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class UploadChartSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        UploadChartRequest request = new UploadChartRequest();
        UploadChartRequestBody bodybody = new UploadChartRequestBody();
        bodybody.withParameters("{\"override\":true,\"skip_lint\":true,\"source\":\"package\"}")
            .withContent("chart-file.tgz");
        body.withBody(bodybody);
        request.withBody(listbodyBody);
        try {
            UploadChartResponse response = client.uploadChart(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

```
}  
}
```

Python

```
# coding: utf-8  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkcce.v3.region.cce_region import CceRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkcce.v3 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = __import__('os').getenv("CLOUD_SDK_AK")  
    sk = __import__('os').getenv("CLOUD_SDK_SK")  
  
    credentials = BasicCredentials(ak, sk) \  
  
    client = CceClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(CceRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = UploadChartRequest()  
        bodybody = UploadChartRequestBody(  
            parameters={"override":true,"skip_lint":true,"source":"package"},  
            content="chart-file.tgz"  
        )  
        request.body = listBodybody  
        response = client.upload_chart(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

Go

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        Build()
```

```
client := cce.NewCceClient(  
    cce.CceClientBuilder().  
        WithRegion(region.ValueOf("<YOUR REGION>")).  
        WithCredential(auth).  
        Build())  
  
request := &model.UploadChartRequest{}  
parametersBody := "{\"override\":true,\"skip_lint\":true,\"source\":\"package\"}"  
bodybody := &model.UploadChartRequestBody{  
    Parameters: &parametersBody,  
    Content: "chart-file.tgz",  
}  
request.Body = listBodybody  
response, err := client.UploadChart(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
201	Created

Error Codes

See [Error Codes](#).

4.12.2 Obtaining a Chart List

Function

This API is used to obtain the chart list.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/charts

Request Parameters

Table 4-968 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-969 Response body parameters

Parameter	Type	Description
[items]	Array of ChartResp objects	Chart list

Table 4-970 ChartResp

Parameter	Type	Description
id	String	Chart ID
name	String	Chart name
values	String	Chart value
translate	String	Chart translation resources
instruction	String	Chart description
version	String	Chart version
description	String	Chart description
source	String	Chart source
icon_url	String	URL to chart icons
public	Boolean	Whether the chart is public

Parameter	Type	Description
chart_url	String	URL to the chart
create_at	String	Created at
update_at	String	Updated at

Example Requests

None

Example Responses

Status code: 200

OK

```
[ {
  "id" : "1abd3bd6-0258-11ec-b8b0-0255ac100b05",
  "name" : "magento-mysql",
  "values" : "{\basic\":{\admin_password\":\*****\",admin_username\":\username\",app_name
\":\magento\",mysql_database\":\magento\",mysql_name\":\mysql\",mysql_password\":\*****
\",mysql_port\":3306,mysql_root_password\":\*****\",mysql_user\":\magento\",storage_class\":\csi-
nas\",storage_mode\":\ReadWriteMany\",storage_size\":\10G\",global\":{\magento_EIP
\":\100.100.100.100\",magento_EPORT\":32080,namespace\":\default\",image\":{\magento_image
\":\example.com/everest/magento:latest\",mysql_image\":\example.com/everest/mysql:5.7.14\"}}",
  "translate" : "",
  "instruction" : "",
  "version" : "1.0.0",
  "description" : "chart description",
  "source" : "",
  "icon_url" : "https://example.com/magento-stack-110x117.png",
  "public" : false,
  "chart_url" : "magento-mysql-1.0.0.tgz",
  "create_at" : "2021-08-20T08:00:29Z",
  "update_at" : "2021-08-20T08:00:29Z"
}]
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ListChartsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
```

security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.

// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment

```
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");

ICredential auth = new BasicCredentials()
    .withAk(ak)
    .withSk(sk);

CceClient client = CceClient.newBuilder()
    .withCredential(auth)
    .withRegion(CceRegion.valueOf("<YOUR REGION>"))
    .build();
ListChartsRequest request = new ListChartsRequest();
try {
    ListChartsResponse response = client.listCharts(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *
```

```
if __name__ == "__main__":
```

```
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
```

```
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
```

```
    credentials = BasicCredentials(ak, sk) \
```

```
    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()
```

```
    try:
        request = ListChartsRequest()
        response = client.list_charts(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```


Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListChartsRequest{}
    response, err := client.ListCharts(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	OK

Error Codes

See [Error Codes](#).

4.12.3 Obtaining a Release List

Function

This API is used to obtain a release list.

Calling Method

For details, see [Calling APIs](#).

URI

GET /cce/cam/v3/clusters/{cluster_id}/releases

Table 4-971 Path Parameters

Parameter	Mandatory	Type	Description
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Table 4-972 Query Parameters

Parameter	Mandatory	Type	Description
chart_id	No	String	Chart ID
namespace	No	String	Namespace of the chart

Request Parameters

Table 4-973 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-974 Response body parameters

Parameter	Type	Description
[items]	Array<Array< ReleaseResp >>	OK

Table 4-975 ReleaseResp

Parameter	Type	Description
chart_name	String	Chart name
chart_public	Boolean	Whether the chart is public
chart_version	String	Chart version
cluster_id	String	Cluster ID
cluster_name	String	Current name
create_at	String	Created at
description	String	Release description
name	String	Release name
namespace	String	Namespace to which a chart release belongs
parameters	String	Release parameters
resources	String	Resources required by the release
status	String	Release status
status_description	String	Release status description
update_at	String	Updated at
values	String	Release value
version	Integer	Release version

Example Requests

None

Example Responses

Status code: 200

OK

```
[ {
  "chart_name": "magento-mysql",
  "chart_public": false,
  "chart_version": "1.0.0",
  "cluster_id": "a870253f-5dc7-11ee-bf71-0255ac100b03",
  "cluster_name": "sfs-turbo-test",
  "create_at": "2023-11-14T20:30:57+08:00",
  "description": "Initial install underway",
  "name": "testwww",
  "namespace": "monitoring",
  "parameters": "",
  "resources": "",
  "status": "PENDING_INSTALL",
  "status_description": "Initial install underway",
  "update_at": "2023-11-14T20:30:57+08:00",
  "values": "{\n  \"basic\": {\n    \"admin_password\": \"*****\",\n    \"admin_username\": \"username\",\n    \"app_name\": \"magento\",\n    \"mysql_database\": \"magento\",\n    \"mysql_name\": \"mysql\",\n    \"mysql_password\": \"*****\",\n    \"mysql_port\": 3306,\n    \"mysql_root_password\": \"*****\",\n    \"mysql_user\": \"magento\",\n    \"storage_class\": \"csi-nas\",\n    \"storage_mode\": \"ReadWriteMany\",\n    \"storage_size\": \"10G\",\n    \"global\": {\n      \"magento_EIP\": \"100.100.100.100\",\n      \"magento_EPORT\": 32080,\n      \"namespace\": \"default\",\n      \"image\": {\n        \"magento_image\": \"example.com/everest/magento:latest\",\n        \"mysql_image\": \"example.com/everest/mysql:5.7.14\"}\n    }\n  },\n  \"version\": 1\n}"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ListReleasesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
    }
}
```

```
ListReleasesRequest request = new ListReleasesRequest();
request.withClusterId("{cluster_id}");
try {
    ListReleasesResponse response = client.listReleases(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListReleasesRequest()
        request.cluster_id = "{cluster_id}"
        response = client.list_releases(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
```

```

risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    Build()

client := cce.NewCceClient(
    cce.CceClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListReleasesRequest{}
request.ClusterId = "{cluster_id}"
response, err := client.ListReleases(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	OK

Error Codes

See [Error Codes](#).

4.12.4 Updating a Chart

Function

This API is used to update a chart.

Calling Method

For details, see [Calling APIs](#).

URI

PUT /v2/charts/{chart_id}

Table 4-976 Path Parameters

Parameter	Mandatory	Type	Description
chart_id	Yes	String	Chart ID

Request Parameters

Table 4-977 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Table 4-978 FormData parameters

Parameter	Mandatory	Type	Description
parameters	No	String	Parameter configurations for uploading a chart. The following is an example: <pre>"{"override":true,"skip_lint":true,"source":"package"}"-</pre> skip_lint: whether lint uploaded chart - override: whether override existed chart - visible: update chart visible
content	Yes	File	Chart package file

Response Parameters

Status code: 200

Table 4-979 Response body parameters

Parameter	Type	Description
id	String	Chart ID

Parameter	Type	Description
name	String	Chart name
values	String	Chart value
translate	String	Chart translation resources
instruction	String	Chart description
version	String	Chart version
description	String	Chart description
source	String	Chart source
icon_url	String	URL to chart icons
public	Boolean	Whether the chart is public
chart_url	String	URL to the chart
create_at	String	Created at
update_at	String	Updated at

Example Requests

```
PUT /v2/charts/{chart_id}

{
  "parameters" : "{ \"override\":true, \"skip_lint\":true, \"source\":\"package\" }",
  "content" : "chart-file.tgz"
}
```

Example Responses

Status code: 200

OK

```
{
  "id" : "e99a7e86-afdd-11eb-aca3-0255ac100b0e",
  "name" : "neo4j",
  "values" : "{ \"acceptLicenseAgreement\": \"no\", \"affinity\": {}, \"authEnabled\": true, \"clusterDomain\": \"cluster.local\", \"core\": { \"initContainers\": [], \"numberOfServers\": 3, \"persistentVolume\": { \"enabled\": true, \"mountPath\": \"/data\", \"size\": \"10Gi\" }, \"sidecarContainers\": [], \"defaultDatabase\": \"neo4j\", \"image\": \"neo4j\", \"imagePullPolicy\": \"IfNotPresent\", \"imageTag\": \"4.0.3-enterprise\", \"name\": \"neo4j\", \"nodeSelector\": {}, \"podDisruptionBudget\": {}, \"readReplica\": { \"autoscaling\": { \"enabled\": false, \"maxReplicas\": 3, \"minReplicas\": 1, \"targetAverageUtilization\": 70 }, \"initContainers\": [], \"numberOfServers\": 0, \"resources\": {}, \"sidecarContainers\": [], \"resources\": {}, \"testImage\": \"markhneedham/k8s-kubectl\", \"testImageTag\": \"master\", \"tolerations\": [], \"useAPOC\": \"true\" }",
  "translate" : "",
  "instruction" : "README.md",
  "version" : "3.0.1",
  "description" : "DEPRECATED Neo4j is the world's leading graph database",
  "source" : "",
  "icon_url" : "https://example.com/images/neo4j_logo.png",
  "public" : false,
  "chart_url" : "neo4j-3.0.1.tgz",
  "create_at" : "2021-05-08T08:53:13Z",
```



```
"update_at" : "2021-05-08T08:53:13Z"
}
```

Status Codes

Status Code	Description
200	OK

Error Codes

See [Error Codes](#).

4.12.5 Creating a Release

Function

This API is used to create a release.

Calling Method

For details, see [Calling APIs](#).

URI

POST /cce/cam/v3/clusters/{cluster_id}/releases

Table 4-980 Path Parameters

Parameter	Mandatory	Type	Description
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-981 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Table 4-982 Request body parameters

Parameter	Mandatory	Type	Description
chart_id	Yes	String	Chart ID
description	No	String	Release description
name	Yes	String	Release name
namespace	Yes	String	Namespace to which a chart release belongs
version	Yes	String	Release version
parameters	No	ReleaseReqBodyParams object	Release parameters
values	Yes	values object	Release value

Table 4-983 ReleaseReqBodyParams

Parameter	Mandatory	Type	Description
dry_run	No	Boolean	After this function is enabled, only chart parameters are verified, and installation is not performed.
name_template	No	String	Release name template
no_hooks	No	Boolean	Whether to disable hooks during installation
replace	No	Boolean	Whether to replace the release with the same name
recreate	No	Boolean	Whether to rebuild the release

Parameter	Mandatory	Type	Description
reset_values	No	Boolean	Whether to reset values during an update
release_version	No	Integer	Version of the rollback release
include_hooks	No	Boolean	Enable hooks during an update or deletion.

Table 4-984 values

Parameter	Mandatory	Type	Description
imagePullPolicy	No	String	Image pull policy
imageTag	No	String	Image tag

Response Parameters

Status code: 201

Table 4-985 Response body parameters

Parameter	Type	Description
chart_name	String	Chart name
chart_public	Boolean	Whether the chart is public
chart_version	String	Chart version
cluster_id	String	Cluster ID
cluster_name	String	Current name
create_at	String	Created at
description	String	Release description
name	String	Release name
namespace	String	Namespace to which a chart release belongs
parameters	String	Release parameters
resources	String	Resources required by the release
status	String	Release status
status_description	String	Release status description

Parameter	Type	Description
update_at	String	Updated at
values	String	Release value
version	Integer	Release version

Example Requests

```
POST /cce/cam/v3/clusters/{cluster_id}/releases
{
  "name": "nino21",
  "namespace": "project01",
  "version": "1.0.0",
  "chart_id": "3c138b72-7ce4-6d76-7c55-604cdb2ce423",
  "values": {
    "imageTag": "v2",
    "imagePullPolicy": "IfNotPresent"
  },
  "parameters": {
    "dry_run": false,
    "no_hooks": false,
    "replace": false,
    "name_template": ""
  }
}
```

Example Responses

Status code: 201

Created

```
{
  "chart_name": "magento-mysql",
  "chart_public": false,
  "chart_version": "1.0.0",
  "cluster_id": "a870253f-5dc7-11ee-bf71-0255ac100b03",
  "cluster_name": "sfs-turbo-test",
  "create_at": "2023-11-14T20:30:57+08:00",
  "description": "Initial install underway",
  "name": "testwww",
  "namespace": "monitoring",
  "parameters": "",
  "resources": "",
  "status": "PENDING_INSTALL",
  "status_description": "Initial install underway",
  "update_at": "2023-11-14T20:30:57+08:00",
  "values": "{\"basic\":{\"admin_password\":\"*****\",\"admin_username\":\"username\",\"app_name\":\"magento\",\"mysql_database\":\"magento\",\"mysql_name\":\"mysql\",\"mysql_password\":\"*****\",\"mysql_port\":\"3306\",\"mysql_root_password\":\"*****\",\"mysql_user\":\"magento\",\"storage_class\":\"csi-nas\",\"storage_mode\":\"ReadWriteMany\",\"storage_size\":\"10G\"},\"global\":{\"magento_EIP\":\"100.100.100.100\",\"magento_EPORT\":\"32080\",\"namespace\":\"default\"},\"image\":{\"magento_image\":\"example.com/everest/magento:latest\",\"mysql_image\":\"example.com/everest/mysql:5.7.14\"}}",
  "version": 1
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class CreateReleaseSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateReleaseRequest request = new CreateReleaseRequest();
        request.withClusterId("{cluster_id}");
        CreateReleaseReqBody body = new CreateReleaseReqBody();
        CreateReleaseReqBodyValues valuesbody = new CreateReleaseReqBodyValues();
        valuesbody.withImagePullPolicy("IfNotPresent")
            .withImageTag("v2");
        ReleaseReqBodyParams parametersbody = new ReleaseReqBodyParams();
        parametersbody.withDryRun(false)
            .withNameTemplate("")
            .withNoHooks(false)
            .withReplace(false);
        body.withValues(valuesbody);
        body.withParameters(parametersbody);
        body.withVersion("1.0.0");
        body.withNamespace("project01");
        body.withName("nino21");
        body.withChartId("3c138b72-7ce4-6d76-7c55-604cdb2ce423");
        request.withBody(body);
        try {
            CreateReleaseResponse response = client.createRelease(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateReleaseRequest()
        request.cluster_id = "{cluster_id}"
        valuesbody = CreateReleaseReqBodyValues(
            image_pull_policy="IfNotPresent",
            image_tag="v2"
        )
        parametersbody = ReleaseReqBodyParams(
            dry_run=False,
            name_template="",
            no_hooks=False,
            replace=False
        )
        request.body = CreateReleaseReqBody(
            values=valuesbody,
            parameters=parametersbody,
            version="1.0.0",
            namespace="project01",
            name="nino21",
            chart_id="3c138b72-7ce4-6d76-7c55-604cdb2ce423"
        )
        response = client.create_release(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
```

```

variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    Build()

client := cce.NewCceClient(
    cce.CceClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateReleaseRequest{}
request.ClusterId = "{cluster_id}"
imagePullPolicyValues:= "IfNotPresent"
imageTagValues:= "v2"
valuesbody := &model.CreateReleaseReqBodyValues{
    ImagePullPolicy: &imagePullPolicyValues,
    ImageTag: &imageTagValues,
}
dryRunParameters:= false
nameTemplateParameters:= ""
noHooksParameters:= false
replaceParameters:= false
parametersbody := &model.ReleaseReqBodyParams{
    DryRun: &dryRunParameters,
    NameTemplate: &nameTemplateParameters,
    NoHooks: &noHooksParameters,
    Replace: &replaceParameters,
}
request.Body = &model.CreateReleaseReqBody{
    Values: valuesbody,
    Parameters: parametersbody,
    Version: "1.0.0",
    Namespace: "project01",
    Name: "nino21",
    ChartId: "3c138b72-7ce4-6d76-7c55-604cdb2ce423",
}
response, err := client.CreateRelease(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
201	Created

Error Codes

See [Error Codes](#).

4.12.6 Deleting a Chart

Function

This API is used to delete a chart.

Calling Method

For details, see [Calling APIs](#).

URI

DELETE /v2/charts/{chart_id}

Table 4-986 Path Parameters

Parameter	Mandatory	Type	Description
chart_id	Yes	String	Chart ID

Request Parameters

Table 4-987 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-988 Response body parameters

Parameter	Type	Description
-	String	

Example Requests

None

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class DeleteChartSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteChartRequest request = new DeleteChartRequest();
        request.withChartId("{chart_id}");
        try {
            DeleteChartResponse response = client.deleteChart(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
        }
    }
}
```

```
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteChartRequest()
        request.chart_id = "{chart_id}"
        response = client.delete_chart(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()
```

```

client := cce.NewCceClient(
    cce.CceClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.DeleteChartRequest{}
request.ChartId = "{chart_id}"
response, err := client.DeleteChart(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	OK

Error Codes

See [Error Codes](#).

4.12.7 Updating a Release

Function

This API is used to update a release.

Calling Method

For details, see [Calling APIs](#).

URI

PUT /cce/cam/v3/clusters/{cluster_id}/namespace/{namespace}/releases/{name}

Table 4-989 Path Parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Release name
namespace	Yes	String	Namespace of the chart

Parameter	Mandatory	Type	Description
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-990 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Table 4-991 Request body parameters

Parameter	Mandatory	Type	Description
chart_id	Yes	String	Chart ID
action	Yes	String	upgrade for an upgrade and rollback for a rollback
parameters	Yes	ReleaseReqBodyParams object	Release parameters
values	Yes	values object	Release value

Table 4-992 ReleaseReqBodyParams

Parameter	Mandatory	Type	Description
dry_run	No	Boolean	After this function is enabled, only chart parameters are verified, and installation is not performed.

Parameter	Mandatory	Type	Description
name_template	No	String	Release name template
no_hooks	No	Boolean	Whether to disable hooks during installation
replace	No	Boolean	Whether to replace the release with the same name
recreate	No	Boolean	Whether to rebuild the release
reset_values	No	Boolean	Whether to reset values during an update
release_version	No	Integer	Version of the rollback release
include_hooks	No	Boolean	Enable hooks during an update or deletion.

Table 4-993 values

Parameter	Mandatory	Type	Description
imagePullPolicy	No	String	Image pull policy
imageTag	No	String	Image tag

Response Parameters

Status code: 200

Table 4-994 Response body parameters

Parameter	Type	Description
chart_name	String	Chart name
chart_public	Boolean	Whether the chart is public
chart_version	String	Chart version
cluster_id	String	Cluster ID
cluster_name	String	Current name
create_at	String	Created at
description	String	Release description

Parameter	Type	Description
name	String	Release name
namespace	String	Namespace to which a chart release belongs
parameters	String	Release parameters
resources	String	Resources required by the release
status	String	Release status
status_description	String	Release status description
update_at	String	Updated at
values	String	Release value
version	Integer	Release version

Example Requests

PUT /cce/cam/v3/clusters/{cluster_id}/namespace/{namespace}/releases/{name}

```
{
  "chart_id" : "af4b699e-018c-11ec-b8b0-0255ac100b05",
  "action" : "upgrade",
  "parameters" : {
    "dry_run" : false,
    "name_template" : "string",
    "no_hooks" : false,
    "replace" : false,
    "recreate" : false,
    "reset_values" : false,
    "release_version" : 1,
    "include_hooks" : false
  },
  "values" : {
    "imagePullPolicy" : "IfNotPresent",
    "imageTag" : "v2"
  }
}
```

Example Responses

Status code: 200

OK

```
{
  "chart_name" : "magento-mysql",
  "chart_public" : false,
  "chart_version" : "1.0.0",
  "cluster_id" : "a870253f-5dc7-11ee-bf71-0255ac100b03",
  "cluster_name" : "sfs-turbo-test",
  "create_at" : "2023-11-14T20:30:57+08:00",
  "description" : "Initial install underway",
  "name" : "testwww",
  "namespace" : "monitoring",
  "parameters" : "",
  "resources" : "",
  "status" : "PENDING_INSTALL",
}
```

```
"status_description" : "Initial install underway",
"update_at" : "2023-11-14T20:30:57+08:00",
"values" : "{ \"basic\": { \"admin_password\": \"*****\", \"admin_username\": \"username\", \"app_name\": \"magento\", \"mysql_database\": \"magento\", \"mysql_name\": \"mysql\", \"mysql_password\": \"*****\", \"mysql_port\": 3306, \"mysql_root_password\": \"*****\", \"mysql_user\": \"magento\", \"storage_class\": \"csi-nas\", \"storage_mode\": \"ReadWriteMany\", \"storage_size\": \"10G\", \"global\": { \"magento_EIP\": \"100.100.100.100\", \"magento_EPORT\": 32080, \"namespace\": \"default\", \"image\": { \"magento_image\": \"example.com/everest/magento:latest\", \"mysql_image\": \"example.com/everest/mysql:5.7.14\" } } }, \"version\" : 1
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class UpdateReleaseSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateReleaseRequest request = new UpdateReleaseRequest();
        request.withName("{name}");
        request.withNamespace("{namespace}");
        request.withClusterId("{cluster_id}");
        UpdateReleaseReqBody body = new UpdateReleaseReqBody();
        UpdateReleaseReqBodyValues valuesbody = new UpdateReleaseReqBodyValues();
        valuesbody.withImagePullPolicy("IfNotPresent")
            .withImageTag("v2");
        ReleaseReqBodyParams parametersbody = new ReleaseReqBodyParams();
        parametersbody.withDryRun(false)
            .withNameTemplate("string")
            .withNoHooks(false)
            .withReplace(false)
            .withRecreate(false)
            .withResetValues(false)
            .withReleaseVersion(1)
            .withIncludeHooks(false);
        body.withValues(valuesbody);
        body.withParameters(parametersbody);
        body.withAction(UpdateReleaseReqBody.ActionEnum.fromValue("upgrade"));
    }
}
```

```
body.withChartId("af4b699e-018c-11ec-b8b0-0255ac100b05");
request.withBody(body);
try {
    UpdateReleaseResponse response = client.updateRelease(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateReleaseRequest()
        request.name = "{name}"
        request.namespace = "{namespace}"
        request.cluster_id = "{cluster_id}"
        valuesbody = UpdateReleaseReqBodyValues(
            image_pull_policy="IfNotPresent",
            image_tag="v2"
        )
        parametersbody = ReleaseReqBodyParams(
            dry_run=False,
            name_template="string",
            no_hooks=False,
            replace=False,
            recreate=False,
            reset_values=False,
            release_version=1,
            include_hooks=False
        )
        request.body = UpdateReleaseReqBody(
            values=valuesbody,
            parameters=parametersbody,
            action="upgrade",
            chart_id="af4b699e-018c-11ec-b8b0-0255ac100b05"
        )
```



```
response = client.update_release(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateReleaseRequest{}
    request.Name = "{name}"
    request.Namespace = "{namespace}"
    request.ClusterId = "{cluster_id}"
    imagePullPolicyValues := "IfNotPresent"
    imageTagValues := "v2"
    valuesbody := &model.UpdateReleaseReqBodyValues{
        ImagePullPolicy: &imagePullPolicyValues,
        ImageTag: &imageTagValues,
    }
    dryRunParameters := false
    nameTemplateParameters := "string"
    noHooksParameters := false
    replaceParameters := false
    recreateParameters := false
    resetValuesParameters := false
    releaseVersionParameters := int32(1)
    includeHooksParameters := false
    parametersbody := &model.ReleaseReqBodyParams{
        DryRun: &dryRunParameters,
        NameTemplate: &nameTemplateParameters,
        NoHooks: &noHooksParameters,
        Replace: &replaceParameters,
        Recreate: &recreateParameters,
        ResetValues: &resetValuesParameters,
        ReleaseVersion: &releaseVersionParameters,
        IncludeHooks: &includeHooksParameters,
    }
    request.Body = &model.UpdateReleaseReqBody{
```

```

Values: valuesbody,
Parameters: parametersbody,
Action: model.GetUpdateReleaseReqBodyActionEnum().UPGRADE,
ChartId: "af4b699e-018c-11ec-b8b0-0255ac100b05",
}
response, err := client.UpdateRelease(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	OK

Error Codes

See [Error Codes](#).

4.12.8 Obtaining a Chart

Function

This API is used to obtain a chart.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/charts/{chart_id}

Table 4-995 Path Parameters

Parameter	Mandatory	Type	Description
chart_id	Yes	String	Chart ID

Request Parameters

Table 4-996 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-997 Response body parameters

Parameter	Type	Description
id	String	Chart ID
name	String	Chart name
values	String	Chart value
translate	String	Chart translation resources
instruction	String	Chart description
version	String	Chart version
description	String	Chart description
source	String	Chart source
icon_url	String	URL to chart icons
public	Boolean	Whether the chart is public
chart_url	String	URL to the chart
create_at	String	Created at
update_at	String	Updated at

Example Requests

None

Example Responses

Status code: 200

OK

```
{
  "id" : "e99a7e86-afdd-11eb-aca3-0255ac100b0e",
  "name" : "neo4j",
  "values" : "{\n  \"acceptLicenseAgreement\" : \"no\",\n  \"affinity\" : {},\n  \"authEnabled\" : true,\n  \"clusterDomain\" : \"cluster.local\",\n  \"core\" : {\n    \"initContainers\" : [],\n    \"numberOfServers\" : 3,\n    \"persistentVolume\" : {\n      \"enabled\" : true,\n      \"mountPath\" : \"/data\",\n      \"size\" : \"10Gi\"\n    },\n    \"sidecarContainers\" : [],\n    \"defaultDatabase\" : \"neo4j\",\n    \"image\" : \"neo4j\",\n    \"imagePullPolicy\" : \"IfNotPresent\",\n    \"imageTag\" : \"4.0.3-enterprise\",\n    \"name\" : \"neo4j\",\n    \"nodeSelector\" : {},\n    \"podDisruptionBudget\" : {},\n    \"readReplica\" : {\n      \"autoscaling\" : {\n        \"enabled\" : false,\n        \"maxReplicas\" : 3,\n        \"minReplicas\" : 1,\n        \"targetAverageUtilization\" : 70\n      },\n      \"initContainers\" : [],\n      \"numberOfServers\" : 0,\n      \"resources\" : {},\n      \"sidecarContainers\" : [],\n      \"resources\" : {},\n      \"testImage\" : \"markhneedham/k8s-kubectl\",\n      \"testImageTag\" : \"master\",\n      \"tolerations\" : [],\n      \"useAPOC\" : \"true\"\n    }\n  },\n  \"translate\" : \"\",\n  \"instruction\" : \"README.md\",\n  \"version\" : \"3.0.1\",\n  \"description\" : \"DEPRECATED Neo4j is the world's leading graph database\",\n  \"source\" : \"\",\n  \"icon_url\" : \"https://info.neo4j.com/rs/773-GON-065/images/neo4j_logo.png\",\n  \"public\" : false,\n  \"chart_url\" : \"neo4j-3.0.1.tgz\",\n  \"create_at\" : \"2021-05-08T08:53:13Z\",\n  \"update_at\" : \"2021-05-08T08:53:13Z\"\n}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ShowChartSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowChartRequest request = new ShowChartRequest();
```

```

request.withChartId("{chart_id}");
try {
    ShowChartResponse response = client.showChart(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
}

```

Python

```

# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowChartRequest()
        request.chart_id = "{chart_id}"
        response = client.show_chart(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)

```

Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment

```

```

variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    Build()

client := cce.NewCceClient(
    cce.CceClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowChartRequest{}
request.ChartId = "{chart_id}"
response, err := client.ShowChart(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	OK

Error Codes

See [Error Codes](#).

4.12.9 Deleting a Release

Function

This API is used to delete a release.

Calling Method

For details, see [Calling APIs](#).

URI

DELETE /cce/cam/v3/clusters/{cluster_id}/namespace/{namespace}/releases/{name}

Table 4-998 Path Parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Release name
namespace	Yes	String	Namespace of the chart
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-999 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-1000 Response body parameters

Parameter	Type	Description
-	String	

Example Requests

None

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class DeleteReleaseSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteReleaseRequest request = new DeleteReleaseRequest();
        request.setName("{name}");
        request.withNamespace("{namespace}");
        request.withClusterId("{cluster_id}");
        try {
            DeleteReleaseResponse response = client.deleteRelease(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
```



```
# The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
variables and decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = BasicCredentials(ak, sk) \

client = CceClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(CceRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = DeleteReleaseRequest()
    request.name = "{name}"
    request.namespace = "{namespace}"
    request.cluster_id = "{cluster_id}"
    response = client.delete_release(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteReleaseRequest{}
    request.Name = "{name}"
    request.Namespace = "{namespace}"
    request.ClusterId = "{cluster_id}"
    response, err := client.DeleteRelease(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

```
}  
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	OK

Error Codes

See [Error Codes](#).

4.12.10 Downloading a Chart

Function

This API is used to download a chart.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/charts/{chart_id}/archive

Table 4-1001 Path Parameters

Parameter	Mandatory	Type	Description
chart_id	Yes	String	Chart ID

Request Parameters

Table 4-1002 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-1003 Response body parameters

Parameter	Type	Description
-	File	

Example Requests

None

Example Responses

Status code: 200

OK

```
"chart-file.tgz"
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class DownloadChartSolution {

    public static void main(String[] args) {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");

ICredential auth = new BasicCredentials()
    .withAk(ak)
    .withSk(sk);

CceClient client = CceClient.newBuilder()
    .withCredential(auth)
    .withRegion(CceRegion.valueOf("<YOUR REGION>"))
    .build();

DownloadChartRequest request = new DownloadChartRequest();
request.withChartId("{chart_id}");
try {
    DownloadChartResponse response = client.downloadChart(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DownloadChartRequest()
        request.chart_id = "{chart_id}"
        response = client.download_chart(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

```
print(e.error_code)
print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DownloadChartRequest{}
    request.ChartId = "{chart_id}"
    response, err := client.DownloadChart(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	OK

Error Codes

See [Error Codes](#).

4.12.11 Obtaining a Release

Function

This API is used to obtain a release.

Calling Method

For details, see [Calling APIs](#).

URI

GET /cce/cam/v3/clusters/{cluster_id}/namespace/{namespace}/releases/{name}

Table 4-1004 Path Parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Release name
namespace	Yes	String	Namespace of the chart
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-1005 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-1006 Response body parameters

Parameter	Type	Description
chart_name	String	Chart name
chart_public	Boolean	Whether the chart is public
chart_version	String	Chart version
cluster_id	String	Cluster ID
cluster_name	String	Current name
create_at	String	Created at
description	String	Release description
name	String	Release name
namespace	String	Namespace to which a chart release belongs
parameters	String	Release parameters
resources	String	Resources required by the release
status	String	Release status
status_description	String	Release status description
update_at	String	Updated at
values	String	Release value
version	Integer	Release version

Example Requests

None

Example Responses

Status code: 200

OK

```
{
  "chart_name": "magento-mysql",
  "chart_public": false,
  "chart_version": "1.0.0",
  "cluster_id": "a870253f-5dc7-11ee-bf71-0255ac100b03",
  "cluster_name": "sfs-turbo-test",
  "create_at": "2023-11-14T20:30:57+08:00",
  "description": "Initial install underway",
  "name": "testwww",
  "namespace": "monitoring",
  "parameters": "",
  "resources": "",
  "status": "PENDING_INSTALL",
```

```
"status_description" : "Initial install underway",
"update_at" : "2023-11-14T20:30:57+08:00",
"values" : "{\"basic\":{\"admin_password\":\"*****\",\"admin_username\":\"username\",\"app_name\":\"magento\",\"mysql_database\":\"magento\",\"mysql_name\":\"mysql\",\"mysql_password\":\"*****\",\"mysql_port\":\"3306\",\"mysql_root_password\":\"*****\",\"mysql_user\":\"magento\",\"storage_class\":\"csi-nas\",\"storage_mode\":\"ReadWriteMany\",\"storage_size\":\"10G\"},\"global\":{\"magento_EIP\":\"100.100.100.100\",\"magento_EPORT\":\"32080\",\"namespace\":\"default\",\"image\":{\"magento_image\":\"example.com/everest/magento:latest\",\"mysql_image\":\"example.com/everest/mysql:5.7.14\"}}\",
"version" : 1
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ShowReleaseSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowReleaseRequest request = new ShowReleaseRequest();
        request.setName("{name}");
        request.withNamespace("{namespace}");
        request.withClusterId("{cluster_id}");
        try {
            ShowReleaseResponse response = client.showRelease(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```


Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowReleaseRequest()
        request.name = "{name}"
        request.namespace = "{namespace}"
        request.cluster_id = "{cluster_id}"
        response = client.show_release(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```

request := &model.ShowReleaseRequest{}
request.Name = "{name}"
request.Namespace = "{namespace}"
request.ClusterId = "{cluster_id}"
response, err := client.ShowRelease(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	OK

Error Codes

See [Error Codes](#).

4.12.12 Obtaining Chart Values

Function

This API is used to obtain chart values.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/charts/{chart_id}/values

Table 4-1007 Path Parameters

Parameter	Mandatory	Type	Description
chart_id	Yes	String	Chart ID

Request Parameters

Table 4-1008 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-1009 Response body parameters

Parameter	Type	Description
values	Map<String,Object>	Data in values.yaml . The data structure depends on the chart.

Example Requests

None

Example Responses

Status code: 200

OK

```
{
  "values": {
    "basic": {
      "admin_password": "*****",
      "admin_username": "username"
    },
    "global": {
      "magento_EIP": "127.0.0.1",
      "magento_EPORT": 32080,
      "namespace": "demo"
    },
    "image": {
      "magento_image": "example.com/demo/magento:latest",
      "mysql_image": "example.com/demo/mysql:5.7.14"
    }
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ShowChartValuesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowChartValuesRequest request = new ShowChartValuesRequest();
        request.withChartId("{chart_id}");
        try {
            ShowChartValuesResponse response = client.showChartValues(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
```

```
variables and decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = BasicCredentials(ak, sk) \

client = CceClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(CceRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ShowChartValuesRequest()
    request.chart_id = "{chart_id}"
    response = client.show_chart_values(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowChartValuesRequest{}
    request.ChartId = "{chart_id}"
    response, err := client.ShowChartValues(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	OK

Error Codes

See [Error Codes](#).

4.12.13 Obtaining Historical Records of a Release

Function

This API is used to obtain historical records of a release.

Calling Method

For details, see [Calling APIs](#).

URI

GET /cce/cam/v3/clusters/{cluster_id}/namespace/{namespace}/releases/{name}/history

Table 4-1010 Path Parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Release name
namespace	Yes	String	Namespace of the chart
cluster_id	Yes	String	Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-1011 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-1012 Response body parameters

Parameter	Type	Description
[items]	Array<Array< ReleaseResp >>	OK

Table 4-1013 ReleaseResp

Parameter	Type	Description
chart_name	String	Chart name
chart_public	Boolean	Whether the chart is public
chart_version	String	Chart version
cluster_id	String	Cluster ID
cluster_name	String	Current name
create_at	String	Created at
description	String	Release description
name	String	Release name
namespace	String	Namespace to which a chart release belongs
parameters	String	Release parameters

Parameter	Type	Description
resources	String	Resources required by the release
status	String	Release status
status_description	String	Release status description
update_at	String	Updated at
values	String	Release value
version	Integer	Release version

Example Requests

None

Example Responses

Status code: 200

OK

```
[ {
  "chart_name": "magento-mysql",
  "chart_public": false,
  "chart_version": "1.0.0",
  "cluster_id": "a870253f-5dc7-11ee-bf71-0255ac100b03",
  "cluster_name": "sfs-turbo-test",
  "create_at": "2023-11-14T20:30:57+08:00",
  "description": "Initial install underway",
  "name": "testwww",
  "namespace": "monitoring",
  "parameters": "",
  "resources": "",
  "status": "PENDING_INSTALL",
  "status_description": "Initial install underway",
  "update_at": "2023-11-14T20:30:57+08:00",
  "values": "{\n  \"basic\": {\n    \"admin_password\": \"*****\",\n    \"admin_username\": \"username\",\n    \"app_name\": \"magento\",\n    \"mysql_database\": \"magento\",\n    \"mysql_name\": \"mysql\",\n    \"mysql_password\": \"*****\",\n    \"mysql_port\": 3306,\n    \"mysql_root_password\": \"*****\",\n    \"mysql_user\": \"magento\",\n    \"storage_class\": \"csinass\",\n    \"storage_mode\": \"ReadWriteMany\",\n    \"storage_size\": \"10G\",\n    \"global\": {\n      \"magento_EIP\": \"100.100.100.100\",\n      \"magento_EPORT\": 32080,\n      \"namespace\": \"default\",\n      \"image\": {\n        \"magento_image\": \"example.com/everest/magento:latest\",\n        \"mysql_image\": \"example.com/everest/mysql:5.7.14\"}}},\n  \"version\": 1\n}"
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
```



```
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ShowReleaseHistorySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();

        ShowReleaseHistoryRequest request = new ShowReleaseHistoryRequest();
        request.withName("{name}");
        request.withNamespace("{namespace}");
        request.withClusterId("{cluster_id}");
        try {
            ShowReleaseHistoryResponse response = client.showReleaseHistory(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
```

```
.with_region(CceRegion.value_of("<YOUR REGION>")) \
.build()

try:
    request = ShowReleaseHistoryRequest()
    request.name = "{name}"
    request.namespace = "{namespace}"
    request.cluster_id = "{cluster_id}"
    response = client.show_release_history(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowReleaseHistoryRequest{}
    request.Name = "{name}"
    request.Namespace = "{namespace}"
    request.ClusterId = "{cluster_id}"
    response, err := client.ShowReleaseHistory(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	OK

Error Codes

See [Error Codes](#).

4.12.14 Obtaining the Quota of a User Chart

Function

This API is used to obtain the quota of a user chart.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/charts/{project_id}/quotas

Table 4-1014 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Request Parameters

Table 4-1015 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format).

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a User Token .

Response Parameters

Status code: 200

Table 4-1016 Response body parameters

Parameter	Type	Description
quotas	quotas object	Chart quota

Table 4-1017 quotas

Parameter	Type	Description
resources	Array of resources objects	Resources

Table 4-1018 resources

Parameter	Type	Description
type	String	Resource type
quota	Integer	Resource quota
used	Integer	Used resources

Example Requests

None

Example Responses

Status code: 200

OK

```
{
  "quotas" : {
    "resources" : [ {
      "type" : "Charts",
      "quota" : 200,
      "used" : 2
    } ]
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cce.v3.region.CceRegion;
import com.huaweicloud.sdk.cce.v3.*;
import com.huaweicloud.sdk.cce.v3.model.*;

public class ShowUserChartsQuotasSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        CceClient client = CceClient.newBuilder()
            .withCredential(auth)
            .withRegion(CceRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowUserChartsQuotasRequest request = new ShowUserChartsQuotasRequest();
        try {
            ShowUserChartsQuotasResponse response = client.showUserChartsQuotas(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcce.v3.region.cce_region import CceRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcce.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = CceClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CceRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowUserChartsQuotasRequest()
        response = client.show_user_charts_quotas(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cce "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cce/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := cce.NewCceClient(
        cce.CceClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```

request := &model.ShowUserChartsQuotasRequest{}
response, err := client.ShowUserChartsQuotas(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	OK

Error Codes

See [Error Codes](#).

4.13 Add-on Instance Parameter Description

4.13.1 virtual-kubelet

Introduction

The virtual-kubelet add-on is an implementation of the open-source Virtual Kubelet project. When momentary traffic spikes occur in CCE clusters, virtual-kubelet uses HUAWEI CLOUD CCI to create pods for Deployments, StatefulSets, and jobs, eliminating the overhead of resizing CCE clusters. For details, see [virtual kubelet](#).

Parameter Description

Table 4-1019 Parameter description

Parameter	Mandatory	Type	Description
basic	Yes	Table 4-1020 object	Basic add-on configuration parameters.
flavor	Yes	Table 4-1021 object	Add-on specifications.

Parameter	Mandatory	Type	Description
custom	Yes	Table 4-1022 object	Custom add-on parameters.

Table 4-1020 basic

Parameter	Mandatory	Type	Description
available_zone	Yes	String	AZ where the add-on is deployed. Example: cn-north-4a
cci_address	Yes	String	Access address and port number of CCI. Example: https://cci.cn-north-4.myhuaweicloud.com:443
euleros_version	Yes	String	EulerOS version. The value is fixed at 2.2.5 .
iam_addresses	Yes	String	Access address and port number of IAM. Example: https://iam.cn-north-4.myhuaweicloud.com:443
swr_addr	Yes	String	Access address of SWR. Example: swr.cn-north-4.myhuaweicloud.com
swr_user	Yes	String	SWR user name. The default value is hwofficial .
rbac_enabled	Yes	String	Whether RBAC is supported. If RBAC is supported, the value is true .
cluster_id	Yes	String	Cluster ID
cluster_name	Yes	String	Cluster name
vpc_id	Yes	String	VPC ID
network_id	Yes	String	Subnet ID
security_group_id	Yes	String	Security group ID
project_id	Yes	String	Project ID
region	Yes	String	Region Example: cn-north-4

Table 4-1021 flavor

Parameter	Mandatory	Type	Description
description	No	String	Add-on description.
name	Yes	String	Add-on flavor name. The value is fixed at Single-instance .
replicas	Yes	String	Number of pods. The default value is 1 .
resources	Yes	resources object	Container resource (CPU and memory) quotas.

Table 4-1022 custom

Parameter	Mandatory	Type	Description
isInstallProxy	No	String	Cross-service interworking. true: Pods in a Cloud Container Engine (CCE) cluster can communicate with pods in a Cloud Container Instance (CCI) cluster through Kubernetes Service.

Table 4-1023 Data structure of the resources field

Parameter	Mandatory	Type	Description
limitsCpu	Yes	String	CPU size limit (unit: MB). Default value: 250m
limitsMem	Yes	String	Memory size limit (unit: Mi). Default value: 512Mi
name	Yes	String	Add-on name. The value is fixed at virtual-kubelet .
requestsCpu	Yes	String	Requested CPU size (unit: MB). Default value: 250m
requestsMem	Yes	String	Requested memory size (unit: Mi). Default value: 512Mi

Example Request

```
{
  "metadata": {
    "annotations": {
      "addon.install/type": "install"
    }
  }
}
```

```

    }
  },
  "spec": {
    "clusterID": "ccbe7bdf-4**9-3**b-b**4-0*****78",
    "version": "1.0.3",
    "addonTemplateName": "virtual-kubelet",
    "values": {
      "basic": {
        "available_zone": "cn-north-4a",
        "cci_address": "https://cci.cn-north-4.myhuaweicloud.com:443",
        "euleros_version": "2.2.5",
        "iam_address": "https://iam.cn-north-4.myhuaweicloud.com:443",
        "swr_addr": "swr.cn-north-4.myhuaweicloud.com",
        "swr_user": "hwofficial",
        "rbac_enabled": true,
        "cluster_id": "ccbe7bdf-4**9-3**b-b**4-0*****78",
        "cluster_name": "test-vk",
        "vpc_id": "2372199f-e6e0-48be-9437-e774aae6bd70",
        "network_id": "863a5e6c-e4f5-45f8-80d9-5090f17a767b",
        "security_group_id": "ac3c1a82-f320-495a-b81d-e7aaa6b304a4",
        "project_id": "085a4*****00a9ccf7fba",
        "region": "cn-north-4"
      },
      "flavor": {
        "description": "Redundancy backup is not available for the add-on.",
        "name": "Single-instance",
        "replicas": 1,
        "resources": [{
          "limitsCpu": "250m",
          "limitsMem": "512Mi",
          "name": "virtual-kubelet",
          "requestsCpu": "250m",
          "requestsMem": "512Mi"
        }]
      },
      "custom": {
        "isInstallProxy": true
      }
    }
  }
}

```

5 Kubernetes APIs

Description

Kubernetes APIs are resource-based (RESTful) programming interfaces provided through HTTP. It supports query, creation, update, and deletion of various cluster resources using standard HTTP request methods (POST, PUT, PATCH, DELETE, and GET).

CCE allows you to use native [Kubernetes APIs](#) in the following ways:

- [Calling Kubernetes APIs Through the Cluster API Server](#). It is suitable for API calls on scale thanks to its direct connection to the API Server. This is a recommended option.
- [Calling Kubernetes APIs Through API Gateway](#). It applies to small-scale API calls. API gateway flow control may be triggered when APIs are called on scale.

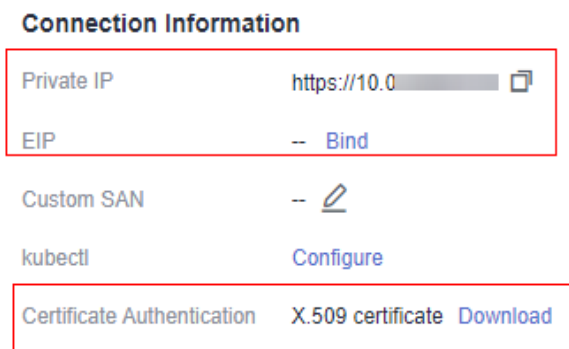
Calling Kubernetes APIs Through the Cluster API Server

You can use the API server of a Kubernetes cluster to call Kubernetes-native APIs.

Step 1 Obtain the cluster certificate and API server.

- Method 1: Obtain the certificate by calling the API for [obtaining the cluster certificate](#), save the returned information to the `kubeconfig.json` file, and extract the certificate, private key, and API server information. The commands are as follows:

```
# Obtain the certificate and save it as client.crt.
cat ./kubeconfig.json |grep client-certificate-data | awk -F "" '{print $4}' | base64 -d > ./client.crt
# Obtain the private key and save it as client.key.
cat ./kubeconfig.json |grep client-key-data | awk -F "" '{print $4}' | base64 -d > ./client.key
# Obtain the API server.
cat ./kubeconfig.json |grep server | awk -F "" '{print $4}'
```
- Method 2: Query the API server address (private or public network address) on the **Cluster Information** page of the CCE console and download the certificate (`client.crt` and `client.key` files).



Step 2 Call Kubernetes-native APIs using the cluster certificate.

For example, run the curl command to call an API to view pod information. In the following example, *192.168.0.198:5443* is the IP address of the cluster API server.

```
curl --cacert ./ca.crt --cert ./client.crt --key ./client.key https://192.168.0.198:5443/api/v1/namespaces/default/pods/
```

For more cluster APIs, see [Kubernetes APIs](#).

----End

Calling Kubernetes APIs Through API Gateway

You can call Kubernetes-native APIs through API Gateway using the URL in the format of **https://{clusterid}.Endpoint/uri**. In the URL, *{clusterid}* indicates the cluster ID, and *uri* indicates the resource path, that is, the path for API access.

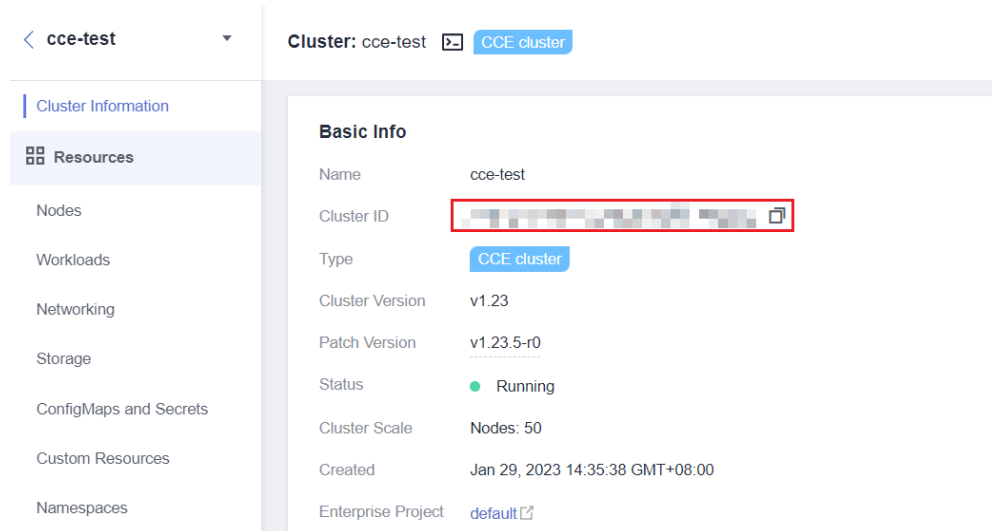
Table 5-1 URL parameters

Parameter	Description
{clusterid}	Cluster ID. After a cluster is created, call the API for obtaining a cluster in a specified project to obtain the cluster ID.
Endpoint	Entry (URL) for a web service, which can be obtained from Endpoints.
uri	Access path of an API for performing an operation. Obtain the value from the URI of the API. For details, see Kubernetes API .

Step 1 Obtain the token of the region where the cluster is located. For details about how to obtain the token, see [Obtaining a Token](#).

Step 2 Obtain the cluster ID using either of the following methods:

- Method 1: Use the API for obtaining cluster information to query the cluster UID.
- Method 2: Query the cluster on the **Cluster Information** page of the CCE console.



Step 3 Determine the requested URL based on the URL format **https://{clusterid}.Endpoint/uri**.

- **{clusterid}**: Obtain the value by using [Step 2](#).
- **Endpoint**: Obtaining the endpoint from [Regions and Endpoints](#).
For example, the endpoint of CCE in the **AP-Singapore** region is **cce.ap-southeast-3.myhuaweicloud.com**.
- **uri**: Set this parameter based on the API to be called. For example, if you want to create a Deployment, the request method is POST and the API URI is **/apis/apps/v1/namespaces/{namespace}/deployments**, where **{namespace}** indicates the cluster namespace name. In this example, the value is **default**.

For more APIs, see [Kubernetes APIs](#).

Combine the preceding parameters following the URL format **https://{clusterid}.Endpoint/uri**.

The following is an example of the URL for calling the API to view information about all pods:

```
https://07da5****.cce.ap-southeast-3.myhuaweicloud.com/apis/apps/v1/namespaces/default/deployments
```

Step 4 Use the request method specified by the API and set the request header parameters. If parameters in the body need to be added, add the structure corresponding to the API by referring to [Kubernetes APIs](#).

Example curl command to call the API for creating a Deployment using POST and adding the corresponding body:

In this example, the **nginx.json** file is used to create a Deployment named **nginx**. The Deployment uses the **nginx:latest** image and contains two pods. Each pod occupies 100m CPU and 200 MiB memory.

```
curl --location --request POST 'https://07da5****.cce.ap-southeast-3.myhuaweicloud.com/apis/apps/v1/namespaces/default/deployments' \
--header 'Content-Type: application/json' \
--header 'X-Auth-Token: MIIWvw****' \
--data @nginx.json
```

Header parameters contained in the request are as follows:

Table 5-2 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format), for example, application/json.
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details, see Obtaining a Token .

The content of the **nginx.json** file is as follows:

```
{
  "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"
          }
        ]
      }
    }
  }
}
```

```
}  
}
```

----End

Related Documents

- [Accessing a Cluster Using Kubernetes APIs](#)
- [Kubernetes official SDKs](#) (including Go, Python, and Java)

Language	Client Library	Sample Program
C	github.com/kubernetes-client/c	Browse
dotnet	github.com/kubernetes-client/csharp	Browse
Go	github.com/kubernetes/client-go/	Browse
Haskell	github.com/kubernetes-client/haskell	Browse
Java	github.com/kubernetes-client/java	Browse
JavaScript	github.com/kubernetes-client/javascript	Browse
Perl	github.com/kubernetes-client/perl/	Browse
Python	github.com/kubernetes-client/python/	Browse
Ruby	github.com/kubernetes-client/ruby/	Browse

6 Out-of-Date APIs

6.1 Obtaining Cluster Certificates

Function

This API is used to obtain certificates of a specified cluster. This API has been deprecated. Use [this API](#) to obtain the cluster certificate instead.

URI

GET /api/v3/projects/{project_id}/clusters/{cluster_id}/clustercert

Table 6-1 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain a project ID, see How to Obtain Parameters in the API URI .
cluster_id	Yes	String	Cluster ID. For details about how to obtain a cluster ID, see How to Obtain Parameters in the API URI .

Request

Table 6-2 Request header parameter description

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format). Possible values: <ul style="list-style-type: none"> • application/json; charset=utf-8 • application/json
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details on how to obtain a user token, see Authentication . Maximum length: 16,384 characters

Response

Status code: 200

Table 6-3 Response body parameters

Parameter	Type	Description
kind	String	API type. The value is fixed at Config and cannot be changed. Default value: Config
apiVersion	String	API version. The value is fixed to v1 .
preferences	String	This field is not used currently and is left unspecified by default.
clusters	Array of Clusters objects	Cluster list.
users	Array of Users objects	Stores the certificate information and client key information of a specified user.
contexts	Array of Contexts objects	Context list.

Parameter	Type	Description
current-context	String	Current context. If publicip (elastic IP address of the VM) exists, the value is external . If publicip does not exist, the value is internal .

Table 6-4 Clusters

Parameter	Type	Description
name	String	Cluster name. <ul style="list-style-type: none"> If the publicip parameter does not exist (that is, no EIP exists), there is only one cluster in the cluster list, and the value of this parameter is internalCluster. If the publicip parameter exists (that is, the EIP exists), there are at least two clusters in the cluster list, and the value of this parameter is externalCluster.
cluster	ClusterCert object	Cluster information.

Table 6-5 ClusterCert

Parameter	Type	Description
server	String	Server IP address.
certificate-authority-data	String	Certificate authorization data.
insecure-skip-tls-verify	Boolean	Whether to skip the server certificate verification. If the cluster type is externalCluster , the value is true .

Table 6-6 Users

Parameter	Type	Description
name	String	The value is fixed to user .
user	User object	Stores the certificate information and client key information of a specified user.

Table 6-7 User

Parameter	Type	Description
client-certificate-data	String	Client certificate.
client-key-data	String	PEM encoding data from the TLS client key file.

Table 6-8 Contexts

Parameter	Type	Description
name	String	Context name. <ul style="list-style-type: none"> If the publicip parameter does not exist (that is, no EIP exists), there is only one cluster in the cluster list, and the value of this parameter is internal. If the publicip parameter exists (that is, the EIP exists), there are at least two clusters in the cluster list, and the value of this parameter is external.
context	Context object	Context information.

Table 6-9 Context

Parameter	Type	Description
cluster	String	Cluster context.
user	String	User context.

Request Example

None

Example Response

Status code: 200

Certificates of the specified cluster are successfully obtained. For details about the certificate file format, see the following structure of **Config** (Kubernetes v1).

```
{
  "kind": "Config",
  "apiVersion": "v1",
  "preferences": { }
```

```

"clusters" : [ {
  "name" : "internalCluster",
  "cluster" : {
    "server" : "https://192.168.1.7:5443",
    "certificate-authority-data" : "Q2VydGlmaWN*****kQgQ0VSVEIGSUNBVEUtLS0tLQo="
  }
}],
"users" : [ {
  "name" : "user",
  "user" : {
    "client-certificate-data" : "LS0tLS1CRUdJTi*****RkIDQVRFLS0tLS0K",
    "client-key-data" : "LS0tLS1CRUdJTiBSU*****BLRVktLS0tLQo="
  }
}],
"contexts" : [ {
  "name" : "internal",
  "context" : {
    "cluster" : "internalCluster",
    "user" : "user"
  }
}],
"current-context" : "internal"
}

```

Status Code

Status code	Description
200	Certificates of the specified cluster are successfully obtained. For details about the certificate file format, see the structure of Config (Kubernetes v1).

Error Codes

For details, see [Error Codes](#).

6.2 Creating a PersistentVolume

Function

This API is used to create a PersistentVolume associated with cloud storage (for example, EVS, SFS, and OBS). This API has been deprecated. Use the corresponding Kubernetes PV API instead.

NOTE

The format of the storage management URL is **https://{clusterid}.Endpoint/uri**. In the URL, *{clusterid}* indicates a cluster ID, and *uri* indicates the resource path, that is, the path for API access. If **https://Endpoint/uri** is used, the **X-Cluster-ID** parameter must be specified in the request header.

URI

POST /api/v1/cloudpersistentvolumes

Request

Table 6-10 Request header parameter description

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format). Possible values: <ul style="list-style-type: none"> • application/json;charset=utf-8 • application/json
X-Auth-Token	Yes	String	Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details on how to obtain a user token, see Authentication . Maximum length: 16,384 characters
X-Cluster-ID	No	String	Cluster ID. This parameter is mandatory when the URL format https://Endpoint/uri is used. For details about how to obtain the value, see How to Obtain Parameters in the API URI .

Table 6-11 Request body parameters

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	API version. The value is fixed to v1 . Default value: v1
kind	Yes	String	API type. The value is fixed to PersistentVolume . Default value: PersistentVolume
metadata	Yes	PersistentVolumeMetadata object	PersistentVolume metadata information.

Parameter	Mandatory	Type	Description
spec	Yes	PersistentVolumeSpec object	PersistentVolume specifications.
status	No	PersistentVolumeStatus object	PersistentVolume status information.

Table 6-12 PersistentVolumeMetadata

Parameter	Mandatory	Type	Description
name	Yes	String	PV name. Enter 1 to 253 characters starting and ending with a letter or digit. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed. PV names must be unique in a namespace.
labels	No	Map<String,String>	<p>PV labels, in the format of key-value pairs.</p> <ul style="list-style-type: none"> • Key: Enter 1 to 63 characters starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain can be prefixed to a key and contain a maximum of 253 characters. Example DNS subdomain: example.com/my-key • Value: The value can be left blank or 1 to 63 characters, starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed in the character string. <p>Example label: "foo": "bar"</p>

Table 6-13 PersistentVolumeSpec

Parameter	Mandatory	Type	Description
flexVolume	Yes	FlexVolume object	FlexVolume storage plug-in of Kubernetes.
persistentVolumeReclaimPolicy	No	String	<p>PV reclaim policy. Possible values:</p> <ul style="list-style-type: none"> • Retain: The Retain reclaim policy allows for manual reclamation of the resource. When the PVC is deleted, the PV still exists and the volume is considered "released". • Recycle: The Recycle reclaim policy performs a basic scrub (rm -rf / thevolume / *) on the volume and makes it available again for a new claim. • Delete: For volume plug-ins that support the Delete reclaim policy, deletion removes both the PV object from Kubernetes, as well as the associated storage asset in the external infrastructure. Volumes that were dynamically provisioned inherit the reclaim policy of their StorageClass, which defaults to Delete.

Parameter	Mandatory	Type	Description
accessModes	Yes	Array of strings	<p>Access mode of the PersistentVolume.</p> <ul style="list-style-type: none"> • ReadWriteOnce: The volume can be mounted as read-write by a single node. <p>NOTE This function is supported only when the cluster version is v1.13.10 and the storage-driver version is 1.0.19.</p> <ul style="list-style-type: none"> • ReadOnlyMany: The volume can be mounted as read-only by many nodes. • ReadWriteMany: The volume can be mounted as read-write by many nodes.

Table 6-14 FlexVolume

Parameter	Mandatory	Type	Description
driver	Yes	String	<p>Name of the FlexVolume storage plug-in. Set this parameter based on the storage volume type.</p> <ul style="list-style-type: none"> • huawei.com/fuxivol (EVS) • huawei.com/fuxinfs (SFS) • huawei.com/fuxiobs (OBS) • huawei.com/fuxiefs (SFS Turbo)

Parameter	Mandatory	Type	Description
fsType	Yes	String	File system type. Set this parameter based on the storage volume type. <ul style="list-style-type: none"> • ext4: EVS volume. For details, see Using EVS Disks as Storage Volumes. • nfs: SFS volume. For details, see Using SFS File Systems as Storage Volumes. • obs: OBS volume. For details, see Using OBS Buckets as Storage Volumes. • efs: SFS Turbo volume. For details, see Using SFS Turbo File Systems as Storage Volumes.
options	Yes	Options object	FlexVolume configuration items.

Table 6-15 Options

Parameter	Mandatory	Type	Description
fsType	Yes	String	File system type. Set this parameter based on the storage volume type. <ul style="list-style-type: none"> • ext4 (EVS) • nfs (SFS) • obs (OBS) • efs (SFS Turbo)
region	Yes	String	Region where the cloud storage is located.
volumeID	Yes	String	UUID of the cloud storage. If the cloud storage is an OBS bucket, set this parameter to the bucket name.

Parameter	Mandatory	Type	Description
storageType	Yes	String	Cloud storage volume type. <ul style="list-style-type: none"> • bs (EVS) • nfs (SFS) • obs (OBS) • efs (SFS Turbo)

Table 6-16 PersistentVolumeStatus

Parameter	Mandatory	Type	Description
accessModes	No	Array of strings	Access mode of the PersistentVolume.
phase	No	String	PV status. Possible values: <ul style="list-style-type: none"> • Available: The volume is not bound to any PVC. • Bound: The PV has been bound to a PVC. • Released: The PVC bound to the PV has been deleted, but the PV has not been made available again for a new claim in the cluster. • Failed: Automatic reclaim of the volume has failed.

Response

Status code: 201

Table 6-17 Response body parameters

Parameter	Type	Description
apiVersion	String	API version. The value is fixed to v1 . Default value: v1
kind	String	API type. The value is fixed to PersistentVolume . Default value: PersistentVolume
metadata	PersistentVolumeMetadata object	PersistentVolume metadata information.

Parameter	Type	Description
spec	PersistentVolumeSpec object	PersistentVolume specifications.
status	PersistentVolumeStatus object	PersistentVolume status information.

Table 6-18 PersistentVolumeMetadata

Parameter	Type	Description
name	String	PV name. Enter 1 to 253 characters starting and ending with a letter or digit. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed. PV names must be unique in a namespace.
labels	Map<String,String>	<p>PV labels, in the format of key-value pairs.</p> <ul style="list-style-type: none"> • Key: Enter 1 to 63 characters starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain can be prefixed to a key and contain a maximum of 253 characters. Example DNS subdomain: example.com/my-key • Value: The value can be left blank or 1 to 63 characters, starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed in the character string. <p>Example label: "foo": "bar"</p>

Table 6-19 PersistentVolumeSpec

Parameter	Type	Description
flexVolume	FlexVolume object	FlexVolume storage plug-in of Kubernetes.

Parameter	Type	Description
persistentVolumeReclaimPolicy	String	<p>PV reclaim policy. Possible values:</p> <ul style="list-style-type: none"> ● Retain: The Retain reclaim policy allows for manual reclamation of the resource. When the PVC is deleted, the PV still exists and the volume is considered "released". ● Recycle: The Recycle reclaim policy performs a basic scrub (<code>rm -rf / thevolume / *</code>) on the volume and makes it available again for a new claim. ● Delete: For volume plug-ins that support the Delete reclaim policy, deletion removes both the PV object from Kubernetes, as well as the associated storage asset in the external infrastructure. Volumes that were dynamically provisioned inherit the reclaim policy of their StorageClass, which defaults to Delete.
accessModes	Array of strings	<p>Access mode of the PersistentVolume.</p> <ul style="list-style-type: none"> ● ReadWriteOnce: The volume can be mounted as read-write by a single node. <p>NOTE This function is supported only when the cluster version is v1.13.10 and the storage-driver version is 1.0.19.</p> <ul style="list-style-type: none"> ● ReadOnlyMany: The volume can be mounted as read-only by many nodes. ● ReadWriteMany: The volume can be mounted as read-write by many nodes.

Table 6-20 FlexVolume

Parameter	Type	Description
driver	String	<p>Name of the FlexVolume storage plug-in. Set this parameter based on the storage volume type.</p> <ul style="list-style-type: none"> ● huawei.com/fuxivol (EVS) ● huawei.com/fuxinfs (SFS) ● huawei.com/fuxiobs (OBS) ● huawei.com/fuxiefs (SFS Turbo)

Parameter	Type	Description
fsType	String	File system type. Set this parameter based on the storage volume type. <ul style="list-style-type: none"> • ext4: EVS volume. For details, see Using EVS Disks as Storage Volumes. • nfs: SFS volume. For details, see Using SFS File Systems as Storage Volumes. • obs: OBS volume. For details, see Using OBS Buckets as Storage Volumes. • efs: SFS Turbo volume. For details, see Using SFS Turbo File Systems as Storage Volumes.
options	Options object	FlexVolume configuration items.

Table 6-21 Options

Parameter	Type	Description
fsType	String	File system type. Set this parameter based on the storage volume type. <ul style="list-style-type: none"> • ext4 (EVS) • nfs (SFS) • obs (OBS) • efs (SFS Turbo)
region	String	Region where the cloud storage is located.
volumeID	String	UUID of the cloud storage. If the cloud storage is an OBS bucket, set this parameter to the bucket name.
storageType	String	Cloud storage volume type. <ul style="list-style-type: none"> • bs (EVS) • nfs (SFS) • obs (OBS) • efs (SFS Turbo)

Table 6-22 PersistentVolumeStatus

Parameter	Type	Description
accessModes	Array of strings	Access mode of the PersistentVolume.

Parameter	Type	Description
phase	String	PV status. Possible values: <ul style="list-style-type: none"> ● Available: The volume is not bound to any PVC. ● Bound: The PV has been bound to a PVC. ● Released: The PVC bound to the PV has been deleted, but the PV has not been made available again for a new claim in the cluster. ● Failed: Automatic reclaim of the volume has failed.

Request Example

- Specifying an EVS volume ID and creating a PersistentVolume:

```
{
  "apiVersion": "v1",
  "kind": "PersistentVolume",
  "metadata": {
    "labels": {
      "name": "pv-test"
    },
    "name": "pv-test"
  },
  "spec": {
    "accessModes": [ "ReadWriteMany" ],
    "flexVolume": {
      "driver": "huawei.com/fuxivol",
      "fsType": "ext4",
      "options": {
        "fsType": "ext4",
        "kubernetes.io/namespace": "default",
        "region": "southchina",
        "volumeID": "76e01b29-08b9-11e8-9ca5-1051722006ec",
        "storageType": "bs"
      }
    },
    "persistentVolumeReclaimPolicy": "Delete"
  }
}
```

- ```
{
 "apiVersion": "v1",
 "kind": "PersistentVolume",
 "metadata": {
 "labels": {
 "name": "pv-test"
 },
 "name": "pv-test"
 },
 "spec": {
 "accessModes": ["ReadWriteMany"],
 "flexVolume": {
 "driver": "huawei.com/fuxivol",
 "fsType": "ext4",
 "options": {
 "fsType": "ext4",
 "region": "southchina",
 "volumeID": "76e01b29-08b9-11e8-9ca5-1051722006ec",
 "storageType": "bs"
 }
 }
 }
}
```

```
 },
 "persistentVolumeReclaimPolicy" : "Delete"
 }
}
```

## Example Response

### Status code: 201

The job for creating a PersistentVolume is successfully delivered.

```
{
 "kind" : "PersistentVolume",
 "apiVersion" : "v1",
 "metadata" : {
 "name" : "pv-test",
 "namespace" : "default",
 "selfLink" : "/api/v1/namespaces/default/persistentvolumes/pv-test",
 "uid" : "e174188f-ff21-11e7-855b-fa163eaf5675",
 "resourceVersion" : "174229",
 "creationTimestamp" : "2018-01-22T03:11:03Z",
 "labels" : {
 "name" : "pv-test"
 },
 "enable" : true
 },
 "spec" : {
 "capacity" : {
 "storage" : "1Gi"
 },
 "accessModes" : ["ReadWriteMany"],
 "flexVolume" : {
 "driver" : "huawei.com/fuxivol",
 "fsType" : "ext4",
 "options" : {
 "fsType" : "ext4",
 "kubernetes.io/namespace" : "default",
 "volumeID" : "0781b22f-4d89-4e9c-b026-80e545cea16c"
 }
 },
 "persistentVolumeReclaimPolicy" : "Delete"
 },
 "status" : {
 "phase" : "Pending"
 }
}
```

## Status Code

| Status Code | Description                                                        |
|-------------|--------------------------------------------------------------------|
| 201         | The job for creating a PersistentVolume is successfully delivered. |

## Error Codes

For details, see [Error Codes](#).

## 6.3 Deleting a PersistentVolume

### Function

This API is used to delete a PersistentVolume from a specified namespace. Using this API, you can determine whether to retain the cloud storage associated with the PersistentVolume. This API has been deprecated. Use the corresponding Kubernetes PV API instead.

#### NOTE

The format of the storage management URL is **https://{clusterid}.Endpoint/uri**. In the URL, *{clusterid}* indicates a cluster ID, and *uri* indicates the resource path, that is, the path for API access. If **https://Endpoint/uri** is used, the **X-Cluster-ID** parameter must be specified in the request header.

### URI

DELETE /api/v1/cloudpersistentvolumes/{name}



**Table 6-23** URI parameter description

| Parameter | Mandatory | Type   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------|-----------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| name      | Yes       | String | <p>Name of the PersistentVolume to be deleted.</p> <p>Name format: <i>Volume name</i>[?deleteVolume=<i>BOOLEAN</i>&amp;storageType=<i>Cloud storage volume type</i>]. The content in square brackets can be omitted. Examples:</p> <ul style="list-style-type: none"> <li>• volume-49f1?deleteVolume=true&amp;storageType=bs</li> <li>• volume-49f1</li> </ul> <p>In the first example:</p> <p><b>deleteVolume:</b> Whether to delete associated backend cloud storage after the PersistentVolume is deleted. The value can be <b>true</b> or <b>false</b>.</p> <p><b>NOTE</b><br/>If the storage volume type is <b>efs</b>, this parameter must be set to <b>false</b>.</p> <p><b>storageType:</b> Cloud storage volume type. This parameter is used together with <b>deleteVolume</b>. That is, <b>deleteVolume</b> and <b>storageType</b> must be configured at the same time.</p> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>• <b>bs:</b> EVS</li> <li>• <b>nfs:</b> SFS</li> <li>• <b>obs:</b> OBS</li> <li>• <b>efs:</b> SFS Turbo</li> </ul> |

## Request

**Table 6-24** Request header parameter description

| Parameter    | Mandatory | Type   | Description                                                                                                                                                                                                                                                                                              |
|--------------|-----------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Content-Type | Yes       | String | Message body type (format).<br>Possible values: <ul style="list-style-type: none"> <li>• <b>application/json;charset=utf-8</b></li> <li>• <b>application/json</b></li> </ul>                                                                                                                             |
| X-Auth-Token | Yes       | String | Requests for calling an API can be authenticated using either a token or AK/SK. If token-based authentication is used, this parameter is mandatory and must be set to a user token. For details on how to obtain a user token, see <a href="#">Authentication</a> .<br>Maximum length: 16,384 characters |
| X-Cluster-ID | No        | String | Cluster ID. This parameter is mandatory when the URL format <b>https://Endpoint/uri</b> is used. For details about how to obtain the value, see <a href="#">How to Obtain Parameters in the API URI</a> .                                                                                                |

## Response

**Status code: 200**

**Table 6-25** Response body parameters

| Parameter  | Type                                            | Description                                                                                  |
|------------|-------------------------------------------------|----------------------------------------------------------------------------------------------|
| apiVersion | String                                          | API version. The value is fixed to <b>v1</b> .<br>Default value: v1                          |
| kind       | String                                          | API type. The value is fixed to <b>PersistentVolume</b> .<br>Default value: PersistentVolume |
| metadata   | <a href="#">PersistentVolumeMetadata</a> object | PersistentVolume metadata information.                                                       |

| Parameter | Type                                          | Description                          |
|-----------|-----------------------------------------------|--------------------------------------|
| spec      | <a href="#">PersistentVolumeSpec</a> object   | PersistentVolume specifications.     |
| status    | <a href="#">PersistentVolumeStatus</a> object | PersistentVolume status information. |

**Table 6-26** PersistentVolumeMetadata

| Parameter | Type               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| name      | String             | PV name. Enter 1 to 253 characters starting and ending with a letter or digit. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed. PV names must be unique in a namespace.                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| labels    | Map<String,String> | <p>PV labels, in the format of key-value pairs.</p> <ul style="list-style-type: none"> <li>• <b>Key:</b> Enter 1 to 63 characters starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. A DNS subdomain can be prefixed to a key and contain a maximum of 253 characters. Example DNS subdomain: example.com/my-key</li> <li>• <b>Value:</b> The value can be left blank or a string of 1 to 63 characters starting with a letter or digit. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed in the character string.</li> </ul> <p>Example label: "foo": "bar"</p> |

**Table 6-27** PersistentVolumeSpec

| Parameter  | Type                              | Description                               |
|------------|-----------------------------------|-------------------------------------------|
| flexVolume | <a href="#">FlexVolume</a> object | FlexVolume storage plug-in of Kubernetes. |

| Parameter                     | Type             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------------------|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| persistentVolumeReclaimPolicy | String           | <p>PV reclaim policy. Possible values:</p> <ul style="list-style-type: none"> <li>● <b>Retain</b>: The Retain reclaim policy allows for manual reclamation of the resource. When the PVC is deleted, the PV still exists and the volume is considered "released".</li> <li>● <b>Recycle</b>: The Recycle reclaim policy performs a basic scrub (<code>rm -rf / thevolume / *</code>) on the volume and makes it available again for a new claim.</li> <li>● <b>Delete</b>: For volume plug-ins that support the Delete reclaim policy, deletion removes both the PV object from Kubernetes, as well as the associated storage asset in the external infrastructure. Volumes that were dynamically provisioned inherit the reclaim policy of their StorageClass, which defaults to <b>Delete</b>.</li> </ul> |
| accessModes                   | Array of strings | <p>Access mode of the PersistentVolume.</p> <ul style="list-style-type: none"> <li>● <b>ReadWriteOnce</b>: The volume can be mounted as read-write by a single node.</li> </ul> <p><b>NOTE</b><br/>This function is supported only when the cluster version is v1.13.10 and the storage-driver version is 1.0.19.</p> <ul style="list-style-type: none"> <li>● <b>ReadOnlyMany</b>: The volume can be mounted as read-only by many nodes.</li> <li>● <b>ReadWriteMany</b>: The volume can be mounted as read-write by many nodes.</li> </ul>                                                                                                                                                                                                                                                                |

**Table 6-28** FlexVolume

| Parameter | Type   | Description                                                                                                                                                                                                                                                                                        |
|-----------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| driver    | String | <p>Name of the FlexVolume storage plug-in. Set this parameter based on the storage volume type.</p> <ul style="list-style-type: none"> <li>● huawei.com/fuxivol (EVS)</li> <li>● huawei.com/fuxinfs (SFS)</li> <li>● huawei.com/fuxiobs (OBS)</li> <li>● huawei.com/fuxiefs (SFS Turbo)</li> </ul> |

| Parameter | Type                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| fsType    | String                | File system type. Set this parameter based on the storage volume type. <ul style="list-style-type: none"> <li>• <b>ext4</b>: EVS volume. For details, see <a href="#">Using EVS Disks as Storage Volumes</a>.</li> <li>• <b>nfs</b>: SFS volume. For details, see <a href="#">Using SFS File Systems as Storage Volumes</a>.</li> <li>• <b>obs</b>: OBS volume. For details, see <a href="#">Using OBS Buckets as Storage Volumes</a>.</li> <li>• <b>efs</b>: SFS Turbo volume. For details, see <a href="#">Using SFS Turbo File Systems as Storage Volumes</a>.</li> </ul> |
| options   | <b>Options</b> object | FlexVolume configuration items.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

**Table 6-29** Options

| Parameter   | Type   | Description                                                                                                                                                                                                |
|-------------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| fsType      | String | File system type. Set this parameter based on the storage volume type. <ul style="list-style-type: none"> <li>• ext4 (EVS)</li> <li>• nfs (SFS)</li> <li>• obs (OBS)</li> <li>• efs (SFS Turbo)</li> </ul> |
| region      | String | Region where the cloud storage is located.                                                                                                                                                                 |
| volumeID    | String | UUID of the cloud storage. If the cloud storage is an OBS bucket, set this parameter to the bucket name.                                                                                                   |
| storageType | String | Cloud storage volume type. <ul style="list-style-type: none"> <li>• bs (EVS)</li> <li>• nfs (SFS)</li> <li>• obs (OBS)</li> <li>• efs (SFS Turbo)</li> </ul>                                               |

**Table 6-30** PersistentVolumeStatus

| Parameter   | Type             | Description                          |
|-------------|------------------|--------------------------------------|
| accessModes | Array of strings | Access mode of the PersistentVolume. |

| Parameter | Type   | Description                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| phase     | String | PV status. Possible values: <ul style="list-style-type: none"> <li>● <b>Available:</b> The volume is not bound to any PVC.</li> <li>● <b>Bound:</b> The PV has been bound to a PVC.</li> <li>● <b>Released:</b> The PVC bound to the PV has been deleted, but the PV has not been made available again for a new claim in the cluster.</li> <li>● <b>Failed:</b> Automatic reclaim of the volume has failed.</li> </ul> |

## Request Example

None

## Example Response

**Status code: 200**

The job for deleting a PersistentVolume is successfully delivered.

```
{
 "kind": "PersistentVolume",
 "apiVersion": "v1",
 "metadata": {
 "name": "pv-test",
 "selfLink": "/api/v1/persistentvolumes/pv-test",
 "uid": "0d93181d-3628-11e7-9093-fa163e8c373b",
 "resourceVersion": "180886",
 "creationTimestamp": "2017-05-11T08:58:51Z",
 "labels": {
 "app": "test"
 }
 },
 "spec": {
 "flexVolume": {
 "driver": "huawei.com/fuxivol",
 "fsType": "ext4",
 "options": {
 "fsType": "ext4",
 "kubernetes.io/namespace": "default",
 "volumeID": "0781b22f-4d89-4e9c-b026-80e545cea16c"
 }
 },
 "capacity": {
 "storage": "1Gi"
 },
 "accessModes": ["ReadWriteMany"],
 "persistentVolumeReclaimPolicy": "Delete"
 },
 "status": {
 "phase": "Available"
 }
}
```

## Status Code

| Status Code | Description                                                        |
|-------------|--------------------------------------------------------------------|
| 200         | The job for deleting a PersistentVolume is successfully delivered. |

## Error Codes

For details, see [Error Codes](#).

# 7 Permissions and Supported Actions

---

You can use Identity and Access Management (IAM) for fine-grained permissions management of your CCE clusters. If your account does not need individual IAM users, you can skip this section.

New IAM users do not have any permissions assigned by default. You need to first add them to one or more groups and attach policies or roles to these groups. Users inherit permissions from the groups to which they are added and can perform specific operations on cloud services based on the assigned permissions. For more information about policy syntax and example policies, see [Permissions Overview](#).

You can grant users permissions by using [roles](#) and [policies](#). Roles are provided by IAM to define service-based permissions that match users' job responsibilities. Policies define API-based permissions for operations on specific resources under certain conditions, allowing for more fine-grained, secure access control of cloud resources.

## NOTE

If you want to allow or deny the access to an API, fine-grained authorization is a good choice.

An account has all of the permissions required to call all APIs, but IAM users must have the required permissions specifically assigned. The required permissions are determined by the actions supported by the API. Only users with the permissions allowing for those actions can call the API successfully. For example, if an IAM user wants to query ECSs using an API, the user must have been granted permissions that allow the `ecs:servers:list` action.

## Supported Actions

CCE provides system-defined policies that can be directly used in IAM. You can also create custom policies to supplement system-defined policies for more refined access control. Operations supported by policies are specific to APIs. The following are common concepts related to policies:

- Permissions: statements in a policy that allow or deny certain operations.
- APIs: REST APIs that can be called by a user who has been granted specific permissions.



- **Actions:** specific operations that are allowed or denied in a custom policy.
- **Dependencies:** actions which a specific action depends on. When allowing an action for a user, you also need to allow any existing action dependencies for that user.
- **IAM projects/Enterprise projects:** the authorization scope of a custom policy. A custom policy can be applied to IAM projects or enterprise projects or both. Policies that contain actions for both IAM and enterprise projects can be used and applied for both IAM and Enterprise Management. Policies that contain actions only for IAM projects can be used and applied to IAM only. For details about the differences between IAM and enterprise management, see [What Are the Differences Between IAM and Enterprise Management?](#)

 **NOTE**

The check mark (√) and cross symbol (x) respectively indicate that an action takes effect or does not take effect for the corresponding type of projects.

CCE supports the following actions in custom policies.

**Table 7-1** Cluster management actions

| Permission                      | API                                                                      | Action             | IAM Project | Enterprise Project |
|---------------------------------|--------------------------------------------------------------------------|--------------------|-------------|--------------------|
| Obtaining clusters in a project | GET /api/v3/projects/{project_id}/clusters                               | cce:cluster:list   | √           | √                  |
| Obtaining a cluster             | GET /api/v3/projects/{project_id}/clusters/{cluster_id}                  | cce:cluster:get    | √           | √                  |
| Creating a cluster              | POST /api/v3/projects/{project_id}/clusters                              | cce:cluster:create | √           | √                  |
| Updating a cluster              | PUT /api/v3/projects/{project_id}/clusters/{cluster_id}                  | cce:cluster:update | √           | √                  |
| Deleting a cluster              | DELETE /api/v3/projects/{project_id}/clusters/{cluster_id}               | cce:cluster:delete | √           | √                  |
| Upgrading a cluster             | POST /api/v2/projects/:projectid/clusters/:clusterid/upgrade             | cce:cluster:update | √           | √                  |
| Waking up a cluster             | POST /api/v3/projects/{project_id}/clusters/{cluster_id}/operation/awake | cce:cluster:start  | √           | √                  |

| Permission                               | API                                                                          | Action             | IAM Project | Enterprise Project |
|------------------------------------------|------------------------------------------------------------------------------|--------------------|-------------|--------------------|
| Hibernating a cluster                    | POST /api/v3/projects/{project_id}/clusters/{cluster_id}/operation/hibernate | cce:cluster:stop   | √           | √                  |
| Changing the specifications of a cluster | POST /api/v2/projects/{project_id}/clusters/:clusterid/resize                | cce:cluster:resize | √           | √                  |
| Obtaining the certificate of a cluster   | POST /api/v3/projects/{project_id}/clusters/{cluster_id}/clustercert         | cce:cluster:get    | √           | √                  |

**Table 7-2** Node management actions

| Permission                       | API                                                                     | Action          | IAM Project | Enterprise Project                                                                                                                                |
|----------------------------------|-------------------------------------------------------------------------|-----------------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| Obtaining all nodes in a cluster | GET /api/v3/projects/{project_id}/clusters/{cluster_id}/nodes           | cce:node:list   | √           | √                                                                                                                                                 |
| Obtaining a node                 | GET /api/v3/projects/{project_id}/clusters/{cluster_id}/nodes/{node_id} | cce:node:get    | √           | √                                                                                                                                                 |
| Creating a node                  | POST /api/v3/projects/{project_id}/clusters/{cluster_id}/nodes          | cce:node:create | √           | √<br><b>NOTE</b><br>If you use enterprise project authorization to create a node, you need to add the global permission of <b>evs:quota:get</b> . |
| Updating a node                  | PUT /api/v3/projects/{project_id}/clusters/{cluster_id}/nodes/{node_id} | cce:node:update | √           | √                                                                                                                                                 |

| Permission      | API                                                                        | Action          | IAM Project | Enterprise Project |
|-----------------|----------------------------------------------------------------------------|-----------------|-------------|--------------------|
| Deleting a node | DELETE /api/v3/projects/{project_id}/clusters/{cluster_id}/nodes/{node_id} | cce:node:delete | √           | √                  |

**Table 7-3** Job management actions

| Permission                        | API                                                                                             | Action         | IAM Project | Enterprise Project |
|-----------------------------------|-------------------------------------------------------------------------------------------------|----------------|-------------|--------------------|
| Obtaining information about a job | GET /api/v3/projects/{project_id}/jobs/{job_id}                                                 | cce:job:get    | √           | √                  |
| Listing all jobs                  | GET /api/v2/projects/{project_id}/jobs                                                          | cce:job:list   | √           | √                  |
| Deleting one or all jobs          | DELETE /api/v2/projects/{project_id}/jobs<br>DELETE /api/v2/projects/{project_id}/jobs/{job_id} | cce:job:delete | √           | √                  |

**Table 7-4** Node pool management actions

| Permission                            | API                                                                             | Action              | IAM Project | Enterprise Project |
|---------------------------------------|---------------------------------------------------------------------------------|---------------------|-------------|--------------------|
| Obtaining all node pools in a cluster | GET /api/v3/projects/{project_id}/clusters/{cluster_id}/nodepools               | cce:nodepool:list   | √           | √                  |
| Obtaining a node pool                 | GET /api/v3/projects/{project_id}/clusters/{cluster_id}/nodepools/{nodepool_id} | cce:nodepool:get    | √           | √                  |
| Creating a node pool                  | POST /api/v3/projects/{project_id}/clusters/{cluster_id}/nodepools              | cce:nodepool:create | √           | √                  |
| Updating a node pool                  | PUT /api/v3/projects/{project_id}/clusters/{cluster_id}/nodepools/{nodepool_id} | cce:nodepool:update | √           | √                  |

| Permission           | API                                                                                | Action              | IAM Project | Enterprise Project |
|----------------------|------------------------------------------------------------------------------------|---------------------|-------------|--------------------|
| Deleting a node pool | DELETE /api/v3/projects/{project_id}/clusters/{cluster_id}/nodepools/{nodepool_id} | cce:nodepool:delete | √           | √                  |

**Table 7-5** Chart management actions

| Permission                          | API                    | Action           | IAM Project | Enterprise Project |
|-------------------------------------|------------------------|------------------|-------------|--------------------|
| Updating a chart                    | PUT /v2/charts/{id}    | cce:chart:update | √           | x                  |
| Uploading a chart                   | POST /v2/charts        | cce:chart:upload | √           | x                  |
| Listing all charts                  | GET /v2/charts         | cce:chart:list   | √           | x                  |
| Obtaining information about a chart | GET /v2/charts/{id}    | cce:chart:get    | √           | x                  |
| Deleting a chart                    | DELETE /v2/charts/{id} | cce:chart:delete | √           | x                  |

**Table 7-6** Release management actions

| Permission                            | API                        | Action             | IAM Project | Enterprise Project |
|---------------------------------------|----------------------------|--------------------|-------------|--------------------|
| Updating a release                    | PUT /v2/releases/{name}    | cce:release:update | √           | √                  |
| Listing all releases                  | GET /v2/releases           | cce:release:list   | √           | √                  |
| Creating a release                    | POST /v2/releases          | cce:release:create | √           | √                  |
| Obtaining information about a release | GET /v2/releases/{name}    | cce:release:get    | √           | √                  |
| Deleting a release                    | DELETE /v2/releases/{name} | cce:release:delete | √           | √                  |

**Table 7-7** Storage management actions

| Permission                       | API                                                                      | Action             | IAM Project | Enterprise Project |
|----------------------------------|--------------------------------------------------------------------------|--------------------|-------------|--------------------|
| Creating a PersistentVolumeClaim | POST /api/v1/namespaces/{namespace}/cloudpersistentvolumeclaims          | cce:storage:create | √           | √                  |
| Deleting a PersistentVolumeClaim | DELETE /api/v1/namespaces/{namespace}/cloudpersistentvolumeclaims/{name} | cce:storage:delete | √           | √                  |
| Listing all volumes              | GET /storage/api/v1/namespaces/{namespace}/listvolumes                   | cce:storage:list   | √           | √                  |

**Table 7-8** Add-on management actions

| Permission                   | API                                                | Action                   | IAM Project | Enterprise Project |
|------------------------------|----------------------------------------------------|--------------------------|-------------|--------------------|
| Creating an add-on instance  | POST /api/v3/addons                                | cce:addonInstance:create | √           | √                  |
| Obtaining an add-on instance | GET /api/v3/addons/{id}?cluster_id={cluster_id}    | cce:addonInstance:get    | √           | √                  |
| Listing all add-on instances | GET /api/v3/addons?cluster_id={cluster_id}         | cce:addonInstance:list   | √           | √                  |
| Deleting an add-on instance  | DELETE /api/v3/addons/{id}?cluster_id={cluster_id} | cce:addonInstance:delete | √           | √                  |
| Updating an add-on instance  | PUT /api/v3/addons/{id}                            | cce:addonInstance:update | √           | √                  |

**Table 7-9** Quota management actions

| Permission              | API                                      | Action        | IAM Project | Enterprise Project |
|-------------------------|------------------------------------------|---------------|-------------|--------------------|
| Obtaining quota details | GET /api/v3/projects/{project_id}/quotas | cce:quota:get | √           | √                  |

# 8 Appendix

## 8.1 Status Code

[Table 8-1](#) describes the status codes.

**Table 8-1** Status code

| Stat<br>us<br>Cod<br>e | Code                          | Description                                                                                                                                                                                                                                      |
|------------------------|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 100                    | Continue                      | The server has received the initial part of the request and the client should continue to send the remaining part.<br>It is issued on a provisional basis while request processing continues. It alerts the client to wait for a final response. |
| 101                    | Switching Protocols           | The requester has asked the server to switch protocols and the server has agreed to do so. The target protocol must be more advanced than the source protocol.<br>For example, the current HTTP protocol is switched to a later version of HTTP. |
| 201                    | Created                       | The request has been fulfilled, resulting in the creation of a new resource.                                                                                                                                                                     |
| 202                    | Accepted                      | The request has been accepted for processing, but the processing has not been completed.                                                                                                                                                         |
| 203                    | Non-Authoritative Information | The server successfully processed the request, but is returning information that may be from another source.                                                                                                                                     |

| Stat<br>us<br>Cod<br>e | Code              | Description                                                                                                                                                                                                 |
|------------------------|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 204                    | NoContent         | The server has successfully processed the request, but does not return any content. The status code is returned in response to an HTTP OPTIONS request.                                                     |
| 205                    | Reset Content     | The server successfully processed the request, but is not returning any content.                                                                                                                            |
| 206                    | Partial Content   | The server has successfully processed a part of the GET request.                                                                                                                                            |
| 300                    | Multiple Choices  | There are multiple options for the requested resource. For example, this code could be used to present a list of resource characteristics and addresses from which the client such as a browser may choose. |
| 301                    | Moved Permanently | This and all future requests should be permanently directed to the given URI indicated in this response.                                                                                                    |
| 302                    | Found             | The requested resource was temporarily moved.                                                                                                                                                               |
| 303                    | See Other         | The response to the request can be found under a different URI, and should be retrieved using a GET or POST method.                                                                                         |
| 304                    | Not Modified      | The requested resource has not been modified. In such case, there is no need to retransmit the resource since the client still has a previously-downloaded copy.                                            |
| 305                    | Use Proxy         | The requested resource is available only through a proxy.                                                                                                                                                   |
| 306                    | Unused            | This HTTP status code is no longer used.                                                                                                                                                                    |
| 400                    | BadRequest        | The request is invalid. The client should modify the request instead of re-initiating it.                                                                                                                   |
| 401                    | Unauthorized      | The authorization information provided by the client is incorrect or invalid.                                                                                                                               |
| 402                    | Payment Required  | This status code is reserved for future use.                                                                                                                                                                |



| Stat<br>us<br>Cod<br>e | Code                          | Description                                                                                                                                                                                                                               |
|------------------------|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 403                    | Forbidden                     | The server has received the request and understood it, but the server is refusing to respond to it.<br>The client should modify the request instead of re-initiating it.                                                                  |
| 404                    | NotFound                      | The requested resource cannot be found.<br>The client should modify the request instead of re-initiating it.                                                                                                                              |
| 405                    | MethodNotAllowed              | A request method is not supported for the requested resource.<br>The client should modify the request instead of re-initiating it.                                                                                                        |
| 406                    | Not Acceptable                | The server cannot fulfill the request based on the content characteristics of the request.                                                                                                                                                |
| 407                    | Proxy Authentication Required | This code is similar to 401, but indicates that the client must first authenticate itself with the proxy.                                                                                                                                 |
| 408                    | Request Time-out              | The server timed out waiting for the request.<br>The client may re-initiate the request without modifications at any later time.                                                                                                          |
| 409                    | Conflict                      | The request cannot be processed due to a conflict.<br>This status code indicates that the resource that the client attempts to create already exists, or the request fails to be processed because of the update of the conflict request. |
| 410                    | Gone                          | The requested resource cannot be found.<br>The status code indicates that the requested resource has been deleted permanently.                                                                                                            |
| 411                    | Length Required               | The server refused to process the request because the request does not specify the length of its content.                                                                                                                                 |
| 412                    | Precondition Failed           | The server does not meet one of the preconditions that the requester puts on the request.                                                                                                                                                 |

| Stat<br>us<br>Cod<br>e | Code                            | Description                                                                                                                                                                                                                                                                                                                                                      |
|------------------------|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 413                    | Request Entity Too Large        | The server refuses to process a request because the request entity is too large. The server may disable the connection to prevent the client from sending requests consecutively. If the server temporarily cannot process the request, the response will contain a Retry-After header field.                                                                    |
| 414                    | Request-URI Too Large           | The URI provided was too long for the server to process.                                                                                                                                                                                                                                                                                                         |
| 415                    | Unsupported Media Type          | The server does not support the media type in the request.                                                                                                                                                                                                                                                                                                       |
| 416                    | Requested range not satisfiable | The requested range is invalid.                                                                                                                                                                                                                                                                                                                                  |
| 417                    | Expectation Failed              | The server fails to meet the requirements of the Expect request-header field.                                                                                                                                                                                                                                                                                    |
| 422                    | UnprocessableEntity             | The request is well-formed but is unable to be processed due to semantic errors.                                                                                                                                                                                                                                                                                 |
| 429                    | TooManyRequests                 | The client sends excessive requests to the server within a given time (exceeding the limit on the access frequency of the client), or the server receives excessive requests within a given time (beyond its processing capability). In this case, the client should repeat requests after the time specified in the Retry-After header of the response expires. |
| 500                    | InternalServerError             | The server is able to receive the request but it could not understand the request.                                                                                                                                                                                                                                                                               |
| 501                    | Not Implemented                 | The server does not support the requested function.                                                                                                                                                                                                                                                                                                              |
| 502                    | Bad Gateway                     | The server acting as a gateway or proxy receives an invalid response from a remote server.                                                                                                                                                                                                                                                                       |
| 503                    | ServiceUnavailable              | The requested service is invalid.<br>The client should modify the request instead of re-initiating it.                                                                                                                                                                                                                                                           |
| 504                    | ServerTimeout                   | The server could not return a timely response. The response will reach the client only if the request carries a timeout parameter.                                                                                                                                                                                                                               |

| Status Code | Code                       | Description                                                                |
|-------------|----------------------------|----------------------------------------------------------------------------|
| 505         | HTTP Version not supported | The server does not support the HTTP protocol version used in the request. |

## 8.2 Error Codes

If an error occurs in API calling, no result is returned. Identify the cause based on the error code of each API. If an error occurs in API calling, HTTP status code 4xx or 5xx is returned. The response body contains the specific error code and information. If you fail to locate the cause of an error, contact customer service and provide the error code, so that we can help you solve the problem as soon as possible.

### Format of an Error Response Body

If an error occurs during API calling, an error code and a message will be displayed. The following shows an error response body.

```
{
 "errorMessage": "The format of message is error",
 "errorCode": "CCE.01400001"
}
```

In the preceding information, **errorCode** is an error code, and **errorMessage** describes the error.

### Error Code Description

If an error code starting with **APIGW** is returned after you call an API, rectify the fault by referring to the instructions provided in [Error Codes](#).

| Status Code | Error Code   | Error Message                | Description                    | Troubleshooting                                                                                                    |
|-------------|--------------|------------------------------|--------------------------------|--------------------------------------------------------------------------------------------------------------------|
| 400         | CCE.01400001 | Invalid request.             | Invalid request body.          | Modify the request body based on the returned message and the CCE API documentation, or contact technical support. |
| 400         | CCE.01400002 | Subnet not found in the VPC. | No subnet is found in the VPC. | Check whether the subnet in the request body is in the corresponding VPC.                                          |

| Status Code | Error Code   | Error Message                           | Description                                               | Troubleshooting                                                           |
|-------------|--------------|-----------------------------------------|-----------------------------------------------------------|---------------------------------------------------------------------------|
| 400         | CCE.01400003 | IPv6 not supported for the subnet.      | The subnet does not support IPv6.                         | Use a subnet that supports IPv6.                                          |
| 400         | CCE.01400004 | No available flavors for master nodes.  | There is no available flavor for the master node.         | Change to another available cluster flavor or contact technical support.  |
| 400         | CCE.01400005 | Container network CIDR blocks conflict. | The container CIDR block conflicts with the existing one. | Check the container CIDR block based on the returned message.             |
| 400         | CCE.01400006 | Content type not supported.             | The content type is invalid.                              | Use the supported content type by referring to the CCE API documentation. |
| 400         | CCE.01400007 | Insufficient cluster quota.             | Insufficient cluster quota.                               | Submit a service ticket to increase the cluster quota.                    |
| 400         | CCE.01400008 | Insufficient server quota               | Insufficient ECS quota.                                   | Submit a service ticket to increase the ECS quota.                        |
| 400         | CCE.01400009 | Insufficient CPU quota.                 | Insufficient ECS CPU quota.                               | Submit a service ticket to increase the ECS CPU quota.                    |
| 400         | CCE.01400010 | Insufficient memory quota.              | Insufficient ECS memory quota.                            | Submit a service ticket to increase the ECS memory quota.                 |
| 400         | CCE.01400011 | Insufficient security group quota.      | Insufficient security group quota.                        | Submit a service ticket to increase the security group quota.             |
| 400         | CCE.01400012 | Insufficient EIP quota.                 | Insufficient EIP quota.                                   | Submit a service ticket to increase the EIP quota.                        |

| Status Code | Error Code   | Error Message                                           | Description                                                 | Troubleshooting                                                                   |
|-------------|--------------|---------------------------------------------------------|-------------------------------------------------------------|-----------------------------------------------------------------------------------|
| 400         | CCE.01400013 | Insufficient volume quota.                              | Insufficient disk quota.                                    | Submit a service ticket to increase the disk quota based on the returned message. |
| 400         | CCE.01400014 | Excessive nodes in the cluster.                         | The number of nodes exceeds the cluster scale.              | Submit a service ticket to change the cluster flavor.                             |
| 400         | CCE.01400015 | Version not supported.                                  | The cluster version is not supported.                       | Create a cluster of the supported version based on the returned message.          |
| 400         | CCE.01400016 | Current cluster type does not support this node flavor. | The current cluster type does not support this node flavor. | Use the correct node flavor based on the returned message.                        |
| 400         | CCE.01400017 | No available container CIDR block found.                | No available container CIDR block is found.                 | Use a correct container CIDR block based on the returned message.                 |
| 400         | CCE.01400018 | This type of OS cannot be created in this CCE version.  | The current CCE version does not support this type of OS.   | Use a supported OS based on the returned message.                                 |
| 400         | CCE.01400019 | Insufficient resource tenant quota.                     | The quota of the resource tenant is insufficient.           | Rectify the fault based on the returned message or contact technical support.     |
| 400         | CCE.01400020 | Insufficient VPC quota.                                 | Insufficient VPC quota.                                     | Rectify the fault based on the returned message or contact technical support.     |
| 400         | CCE.01400021 | No available flavors for nodes.                         | There is no available flavor for the node.                  | Change to another available node flavor or contact technical support.             |

| Status Code | Error Code   | Error Message                        | Description                                         | Troubleshooting                                                                                                    |
|-------------|--------------|--------------------------------------|-----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| 400         | CCE.01400022 | No available node volumes for nodes. | There is no available EVS disk flavor for the node. | Change to another available EVS disk flavor or contact technical support.                                          |
| 400         | CCE.02400001 | Invalid request.                     | Invalid request body.                               | Modify the request body based on the returned message and the CCE API documentation, or contact technical support. |
| 400         | CCE.03400001 | Invalid request.                     | Invalid request body.                               | Modify the request body based on the returned message and the CCE API documentation, or contact technical support. |
| 400         | CCE.03400002 | Missing access key.                  | The access key is missing.                          | Ensure that the version of the installed or upgraded storage add-on is correct, or contact technical support.      |
| 401         | CCE.01401001 | Authorization failed.                | Authentication failed.                              | Rectify the fault based on the returned message or contact technical support.                                      |
| 401         | CCE.02401001 | Authorization failed.                | Authentication failed.                              | Rectify the fault based on the returned message or contact technical support.                                      |
| 401         | CCE.03401001 | Authorization failed.                | Authentication failed.                              | Rectify the fault based on the returned message or contact technical support.                                      |

| Status Code | Error Code   | Error Message                              | Description                                                   | Troubleshooting                                                                         |
|-------------|--------------|--------------------------------------------|---------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| 403         | CCE.01403001 | Forbidden.                                 | Access denied.                                                | Rectify the fault based on the returned message or contact technical support.           |
| 403         | CCE.02403001 | Forbidden.                                 | Access denied.                                                | Rectify the fault based on the returned message or contact technical support.           |
| 403         | CCE.03403001 | Forbidden.                                 | Access denied.                                                | Rectify the fault based on the returned message or contact technical support.           |
| 404         | CCE.01404001 | Resource not found.                        | Resources not found.                                          | Check whether the resource to be accessed has been deleted.                             |
| 404         | CCE.02404001 | Resource not found.                        | Resources not found.                                          | Check whether the resource to be accessed has been deleted.                             |
| 404         | CCE.03404001 | Resource not found.                        | Resources not found.                                          | Check whether the resource to be accessed has been deleted.                             |
| 409         | CCE.01409001 | The resource already exists.               | The resource already exists.                                  | Delete the resource and try again.                                                      |
| 409         | CCE.01409002 | Resource updated with out-of-date version. | An out-of-date version is used to update the target resource. | Ensure that the target resource version is the latest one or contact technical support. |
| 409         | CCE.02409001 | The resource already exists.               | The resource already exists.                                  | Delete the resource and try again.                                                      |
| 409         | CCE.03409001 | Addon instance has installed.              | The add-on instance has been installed.                       | Delete the add-on instance and install it again.                                        |

| Status Code | Error Code   | Error Message                              | Description                                      | Troubleshooting                                                               |
|-------------|--------------|--------------------------------------------|--------------------------------------------------|-------------------------------------------------------------------------------|
| 429         | CCE.01429002 | Resource locked by other requests.         | The resource is locked by another request.       | Rectify the fault based on the returned message or contact technical support. |
| 429         | CCE.02429001 | The throttling threshold has been reached. | The maximum number of requests has been reached. | Reduce the frequency of sending requests or contact technical support.        |
| 500         | CCE.01500001 | Internal error.                            | Internal error.                                  | Rectify the fault based on the returned message or contact technical support. |
| 500         | CCE.02500001 | Internal error.                            | Internal error.                                  | Rectify the fault based on the returned message or contact technical support. |
| 500         | CCE.03500001 | Internal error.                            | Internal error.                                  | Rectify the fault based on the returned message or contact technical support. |

## 8.3 Obtaining a Project ID

### Scenarios

A project ID is required for some URLs when APIs are called. Therefore, obtain a project ID in advance. Two methods are available:

- [Call an API](#)
- [Use the console](#)

### Obtaining the Project ID by Calling an API

You can obtain the project ID by calling the API used to [obtain projects based on specified criteria](#).

The API for obtaining a project ID is **GET https://{Endpoint}/v3/projects**. **{Endpoint}** indicates the endpoint of IAM, which can be obtained from [Endpoints](#). For details about API authentication, see [Authentication](#).

The following is an example response. The value of **id** is the project ID.



```
{
 "projects": [
 {
 "domain_id": "65382450e8f64ac0870cd180d14e684b",
 "is_domain": false,
 "parent_id": "65382450e8f64ac0870cd180d14e684b",
 "name": "project_name",
 "description": "",
 "links": {
 "next": null,
 "previous": null,
 "self": "https://www.example.com/v3/projects/a4a5d4098fb4474fa22cd05f897d6b99"
 },
 "id": "a4a5d4098fb4474fa22cd05f897d6b99",
 "enabled": true
 }
],
 "links": {
 "next": null,
 "previous": null,
 "self": "https://www.example.com/v3/projects"
 }
}
```

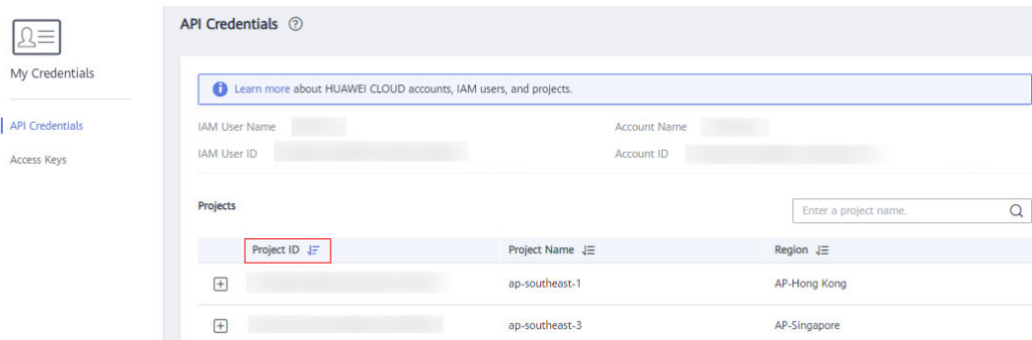
## Obtaining a Project ID from the Console

To obtain a project ID from the console, perform the following operations:

1. Log in to the management console.
2. Hover over the username in the upper right corner and select **My Credentials** from the drop-down list.

On the **API Credentials** page, view the project ID in the project list.

**Figure 8-1** Viewing project IDs



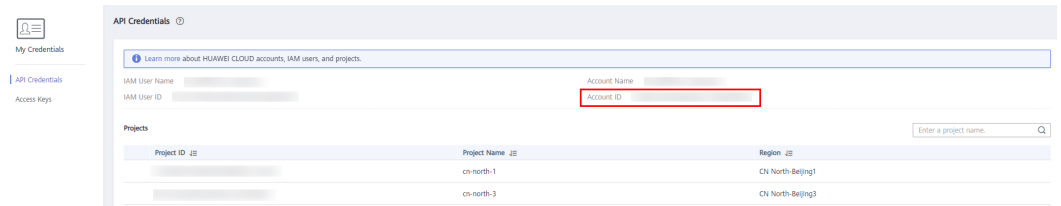
## 8.4 Obtaining an Account ID

An account ID (domain-id) is required for some URLs when an API is called. To obtain an account ID, perform the following operations:

1. Sign up and log in to the management console.
2. Click the username in the upper right corner and choose **My Credentials** from the drop-down list.

On the **API Credentials** page, view the account ID.

Figure 8-2 Obtaining an account ID



## 8.5 Specifying Add-ons to Be Installed During Cluster Creation

When creating a cluster, you can add a key-value pair to **annotations** of the **metadata** field in the request body to install an add-on in the cluster. The key is **cluster.install.addons/install**, and the value is a JSON array of **AddonTemplate**.

Table 8-2 Value data structure

| Parameter | Mandatory | Type                               | Description                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------|-----------|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Value     | Yes       | JSON array of AddonTemplate string | Add-on to be installed in the cluster. If this parameter is not specified, the <b>coredns</b> and <b>storage-driver</b> add-ons are installed by default for clusters of Kubernetes 1.13 and earlier versions, and the <b>coredns</b> , <b>everest</b> , and <b>npd</b> add-ons are installed by default for clusters of Kubernetes 1.15 and later versions. For details, see <a href="#">Table 8-3</a> . |

Table 8-3 Data structure of the AddonTemplate field

| Parameter         | Mandatory | Type   | Description                                                                                                        |
|-------------------|-----------|--------|--------------------------------------------------------------------------------------------------------------------|
| addonTemplateName | Yes       | String | Add-on name.<br>The value can be <b>coredns</b> , <b>autoscaler</b> , <b>gpu-beta</b> , or <b>storage-driver</b> . |

| Parameter | Mandatory | Type     | Description                                                                                                                                                                                                                                                                                                                                                                            |
|-----------|-----------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| version   | No        | String   | Add-on version.<br>To view the add-on version information, log in to the CCE console, click <b>Add-ons</b> in the navigation pane. Click the target add-on name. On the <b>Version History</b> tab page, you can view the versions of the add-on. If this parameter is left blank, the latest version is used by default.                                                              |
| values    | No        | Json Map | Parameters to be set for installing an add-on. <ul style="list-style-type: none"> <li>coredns: For details, see <a href="#">Table 8-4</a>.</li> <li>autoscaler: For details, see <a href="#">Table 8-8</a>.</li> <li>gpu-beta: For details, see <a href="#">Table 8-10</a>.</li> </ul> <b>NOTE</b><br>You do not need to set this parameter when installing the storage-driver add-on. |

**Table 8-4** Special key-value pairs of the coredns values parameters

| Parameter | Mandatory | Type                 | Description                |
|-----------|-----------|----------------------|----------------------------|
| flavor    | Yes       | <b>flavor</b> object | Add-on specifications.     |
| custom    | Yes       | <b>custom</b> object | Custom coredns parameters. |

**Table 8-5** Data structure of the flavor field

| Parameter   | Mandatory | Type                   | Description                |
|-------------|-----------|------------------------|----------------------------|
| description | Yes       | String                 | Specification description. |
| name        | Yes       | String                 | Flavor name.               |
| replicas    | Yes       | Integer                | Number of pods.            |
| resources   | Yes       | <b>resource</b> object | Add-on resources.          |

**Table 8-6** Data structure of the resource field

| Parameter   | Mandatory | Type   | Description         |
|-------------|-----------|--------|---------------------|
| name        | Yes       | String | Resource name.      |
| limitsCpu   | Yes       | String | CPU quota limit.    |
| limitsMem   | Yes       | String | Memory quota limit. |
| requestsCpu | Yes       | String | Requested CPU.      |
| requestsMem | Yes       | String | Requested memory.   |

**Table 8-7** Data structure of the custom field

| Parameter             | Mandatory | Type       | Description                                                                                                                                                                                            |
|-----------------------|-----------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| stub_domains          | No        | Json Map   | JSON map that uses the DNS suffix key (such as acme.local) and the value composed of the JSON array of the DNS IP address.                                                                             |
| upstream_name_servers | No        | Json Array | The value of this parameter replaces the name server value obtained from <code>/etc/resolv.conf</code> of the node by default. Restriction: A maximum of three upstream name servers can be specified. |
| tenant_id             | Yes       | String     | Project ID.                                                                                                                                                                                            |

**Table 8-8** Special key-value pairs of the autoscaler values parameters

| Parameter | Mandatory | Type                 | Description                   |
|-----------|-----------|----------------------|-------------------------------|
| flavor    | Yes       | <b>flavor</b> Object | Add-on specifications.        |
| custom    | Yes       | <b>custom</b> object | Custom autoscaler parameters. |

**Table 8-9** Data structure of the custom field

| Parameter                      | Mandatory | Type    | Description                                                                                                                                                                                                               |
|--------------------------------|-----------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| coresTotal                     | Yes       | Integer | Maximum and minimum number of cores of a cluster. The format is <b>&lt;min&gt;:&lt;max&gt;</b> . The autoscaler add-on performs cluster scaling within this range.                                                        |
| maxEmptyBulkDeleteFlag         | Yes       | Integer | Maximum number of empty nodes that can be deleted at a time.                                                                                                                                                              |
| maxNodesTotal                  | Yes       | Integer | Maximum number of nodes in all node groups. The autoscaler add-on performs cluster scaling within this range.                                                                                                             |
| memoryTotal                    | Yes       | Integer | Maximum and minimum size of memory of a cluster. The format is <b>&lt;min&gt;:&lt;max&gt;</b> . The autoscaler add-on performs cluster scaling within this range.                                                         |
| scaleDownDelayAfterAdd         | Yes       | Integer | Interval for performing scale-down evaluation after a scale-up.                                                                                                                                                           |
| scaleDownDelayAfterDelete      | Yes       | Integer | Interval for performing scale-down evaluation after nodes are deleted. The default value is <b>scanInterval</b> .                                                                                                         |
| scaleDownDelayAfterFailure     | Yes       | Integer | Interval for performing scale-down evaluation after a scale-down failure.                                                                                                                                                 |
| scaleDownEnabled               | Yes       | Boolean | Whether to enable cluster scaling down.                                                                                                                                                                                   |
| scaleDownUnneededTime          | Yes       | Integer | Time waited to start a scale-down after a node has reached the scale-down threshold.                                                                                                                                      |
| scaleDownUtilizationThreshold  | Yes       | Float   | Threshold percentage of the total CPU or memory occupied by all pods running on the node to the allocatable resources on the node. When the actual percentage is lower than this threshold, node scale-down is triggered. |
| scaleUpCpuUtilizationThreshold | Yes       | Float   | CPU usage threshold for triggering scale-up.                                                                                                                                                                              |

| Parameter                      | Mandatory | Type    | Description                                                          |
|--------------------------------|-----------|---------|----------------------------------------------------------------------|
| scaleUpMemUtilizationThreshold | Yes       | Float   | Memory usage threshold for triggering scale-up.                      |
| scaleUpUnscheduledPodEnabled   | Yes       | Boolean | Scale-up is triggered when there are unscheduled pods.               |
| scaleUpUtilizationEnabled      | Yes       | Boolean | Scale-up is triggered when the resource usage reaches the threshold. |
| tenant_id                      | Yes       | String  | Project ID.                                                          |
| unremovableNodeRecheckTimeout  | Yes       | Integer | Interval for checking whether a node can be removed.                 |

**Table 8-10** Special Key-Value in values of AddonTemplate (GPU-beta)

| Parameter | Mandatory | Type                 | Description                      |
|-----------|-----------|----------------------|----------------------------------|
| custom    | Yes       | <b>custom</b> object | Customer parameters of gpu-beta. |

**Table 8-11** Data structure of the custom field

| Parameter                  | Mandatory | Type    | Description                                                           |
|----------------------------|-----------|---------|-----------------------------------------------------------------------|
| is_driver_from_nvidia      | Yes       | Boolean | Whether to use the NVIDIA driver. Set this parameter to <b>true</b> . |
| nvidia_driver_download_url | Yes       | String  | Driver downloading URL.                                               |

**Table 8-12** Data structure of the spec field

| Parameter | Mandatory | Type   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------|-----------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| type      | Yes       | String | <p>Cluster type. Possible values:</p> <ul style="list-style-type: none"> <li> <b>VirtualMachine:</b> The cluster is a hybrid cluster. A hybrid cluster manages a group of node resources based on Kubernetes. It can manage VMs, bare-metal machines, or a combination of both. Kubernetes automatically schedules containers onto available nodes. Before creating a containerized workload, you must have an available cluster. </li> <li> <b>ARM64:</b> The cluster is a Kunpeng cluster. Containers in CCE's Kunpeng clusters can run on Kunpeng servers that use ARM architecture and Kunpeng processors. Kunpeng-accelerated cloud servers are easy to deploy and provide comparable scaling and scheduling performance as x86-based cloud servers. </li> </ul> |

| Parameter | Mandatory | Type   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------|-----------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| flavor    | Yes       | String | <p>Cluster flavor, which cannot be changed after the cluster is created.</p> <ul style="list-style-type: none"><li>• <b>cce.s1.small</b>: a small-scale CCE cluster with one master node and a maximum of 50 worker nodes</li><li>• <b>cce.s1.medium</b>: a medium-scale CCE cluster with one master node and a maximum of 200 worker nodes</li><li>• <b>cce.s2.small</b>: a small-scale, HA CCE cluster with three master nodes and a maximum of 50 worker nodes</li><li>• <b>cce.s2.medium</b>: a medium-scale, HA CCE cluster with three master nodes and a maximum of 200 worker nodes</li><li>• <b>cce.s2.large</b>: a large-scale, HA CCE cluster with three master nodes and a maximum of 1,000 worker nodes</li><li>• <b>cce.s2.xlarge</b>: an ultra-large-scale, HA CCE cluster with three master nodes and a maximum of 2,000 worker nodes</li></ul> |



| Parameter | Mandatory | Type   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------|-----------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|           |           |        | <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>• <b>s1</b>: indicates a cluster with one master node. If the master node is faulty, the cluster is unavailable, but running workloads are not affected.</li> <li>• <b>s2</b>: indicates a cluster with three master nodes. This is an HA cluster. If one of the master nodes is faulty, the cluster is still available.</li> <li>• <b>dec</b>: dedicated hybrid cluster specifications. For example, <b>cce.dec.s1.small</b> is a small-scale, single-master, dedicated hybrid cluster (≤ 50 nodes).</li> <li>• For example, <b>≤ 50 nodes</b> indicates that the maximum number of nodes that can be managed by the cluster is 50.</li> <li>• A single-master cluster has only one master node. If the master node is down, the cluster will become unavailable and stop serving new workloads. However, existing workloads in the cluster are not affected.</li> <li>• An HA cluster has multiple master nodes. Faults in a single master node will not take the cluster down. For details on how to distinguish high-availability clusters from regular clusters, see <a href="#">How Do I Check Whether a Cluster Is in Multi-Master Mode?</a></li> </ul> |
| version   | No        | String | <p>Cluster's baseline Kubernetes version. The latest version is recommended.</p> <ul style="list-style-type: none"> <li>• You can create clusters of two latest versions on the CCE console. To learn which cluster versions are available, choose <b>Clusters</b> in the navigation pane, click <b>Buy Cluster</b>, and view the supported cluster versions.</li> <li>• You can call APIs to create clusters of other versions. However, these clusters will be gradually brought offline. For details about the offline policy, see the official CCE announcement.</li> </ul> <p><b>NOTE</b><br/>If this parameter is left blank, the cluster of the latest version is created by default.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

| Parameter            | Mandatory | Type                                    | Description                                                                                                                                                                                                                                                                                                                           |
|----------------------|-----------|-----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| description          | No        | String                                  | Cluster description, for example, which purpose the cluster is intended to serve. By default, this parameter is left unspecified. To modify cluster description after the cluster is created, call the <a href="#">API used to update information about a specified cluster</a> or go to the cluster details page on the CCE console. |
| ipv6enable           | No        | Boolean                                 | Whether the cluster supports IPv6. Clusters of version 1.15 and later support IPv6.                                                                                                                                                                                                                                                   |
| hostNetwork          | Yes       | <a href="#">HostNetwork</a> object      | Node networking parameters, including VPC and subnet ID. <b>hostNetwork</b> is mandatory because nodes in a cluster communicate with each other by using a VPC.                                                                                                                                                                       |
| containerNetwork     | Yes       | <a href="#">ContainerNetwork</a> object | Container networking parameters, including the container network model and container CIDR block.                                                                                                                                                                                                                                      |
| authentication       | No        | <a href="#">Authentication</a> object   | Configurations of the cluster authentication mode.                                                                                                                                                                                                                                                                                    |
| kubernetesSvcIpRange | No        | String                                  | Service CIDR block or the IP address range which the <b>kubernetes clusterIp</b> must fall within. This parameter is available only for clusters of v1.11.7 and later.                                                                                                                                                                |
| billingMode          | No        | Integer                                 | Billing mode of a cluster. Currently, only <b>pay-per-use</b> clusters can be created.<br>Value <b>0</b> indicates pay-per-use. If this parameter is left unspecified, the default value <b>0</b> is used.                                                                                                                            |
| masters              | No        | <a href="#">MasterSpec</a> object       | Advanced configurations of the master node                                                                                                                                                                                                                                                                                            |
| extendParam          | No        | Map<String, String>                     | Extended fields in the format of key-value pairs.<br>You can configure multi-AZ clusters and dedicated hybrid clusters, and create clusters in specific enterprise projects.                                                                                                                                                          |

## Request Example

```
{
 "kind": "Cluster",
 "apiVersion": "v3",
 "metadata": {
 "name": "mycluster",
 "labels": {
 "foo": "bar"
 },
 "annotations": {
 "cluster.install.addons/install": "[{"addonTemplateName":"autoscaler","version":"1.15.11","values":{"flavor":{"description":"Has only one instance"},"name":"Single","replicas":1,"resources":[{"limitsCpu":"90m","limitsMem":"200Mi","name":"autoscaler","requestsCpu":"50m","requestsMem":"100Mi"}]},"custom":{"coresTotal":32000,"maxEmptyBulkDeleteFlag":10,"maxNodesTotal":1000,"memoryTotal":128000,"scaleDownDelayAfterAdd":10,"scaleDownDelayAfterDelete":10,"scaleDownDelayAfterFailure":3,"scaleDownEnabled":false,"scaleDownUnneededTime":10,"scaleDownUtilizationThreshold":0.5,"scaleUpCpuUtilizationThreshold":1,"scaleUpMemUtilizationThreshold":1,"scaleUpUnscheduledPodEnabled":true,"scaleUpUtilizationEnabled":true,"tenant_id":"22a8a02394794b908d256103a5b63c65"},"unremovableNodeRecheckTimeout":5}]]], [{"addonTemplateName":"coredns","version":"1.15.3","values":{"flavor":{"description":"Has two instances"},"name":"2500","replicas":2,"resources":[{"limitsCpu":"500m","limitsMem":"512Mi","name":"coredns","requestsCpu":"500m","requestsMem":"512Mi"}]},"custom":{"stub_domains":{"upstream_nameservers":[]},"tenant_id":"22a8a02394794b908d256103a5b63c65"}]}], [{"addonTemplateName":"gpu-beta","version":"1.1.7","values":{"custom":{"is_driver_from_nvidia":true,"nvidia_driver_download_url":"https://us.download.nvidia.com/tesla/396.37/NVIDIA-Linux-x86_64-396.37.run"}]}], [{"addonTemplateName":"storage-driver","version":""}]]"
 },
 "spec": {
 "type": "VirtualMachine",
 "flavor": "cce.s1.small",
 "version": "v1.15.11",
 "description": "this is a demo cluster",
 "hostNetwork": {
 "vpc": "1cb74d47-8e09-4d14-a065-75f4fc03e5eb",
 "subnet": "d44c089c-fbdc-49b3-ae8f-7b42c64219cf"
 },
 "containerNetwork": {
 "mode": "overlay_l2",
 "cidr": "172.16.0.0/16"
 },
 "extendParam": {
 "foo": "bar"
 }
 }
}
```

## 8.6 How to Obtain Parameters in the API URI

### project\_id

**project\_id** indicates the project ID, which can be obtained from the console or APIs. For details, see [Obtaining a Project ID](#).

### cluster\_id

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

**Step 2** Click the name of the created cluster. The cluster details page is displayed. Obtain the cluster ID.

**Figure 8-3** Obtaining the cluster ID

**Basic Info**

|                    |                                 |
|--------------------|---------------------------------|
| Name               | cce-test                        |
| Cluster ID         | [Redacted]                      |
| Type               | CCE cluster                     |
| Cluster Version    | v1.23                           |
| Patch Version      | v1.23.5-r0                      |
| Status             | ● Running                       |
| Cluster Scale      | Nodes: 50                       |
| Created            | Jan 29, 2023 14:35:38 GMT+08:00 |
| Enterprise Project | default                         |

----End

**node\_id**

- Step 1** Log in to the CCE console. In the navigation pane, choose **Clusters**.
- Step 2** Click the name of the created cluster, select **Nodes** on the left, and move the cursor to the node name to view the node ID.

**Figure 8-4** Obtaining the node ID

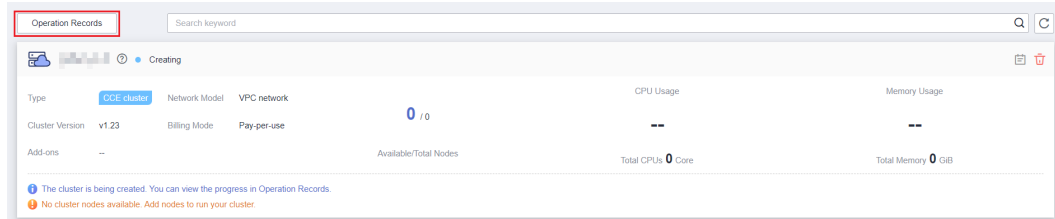


----End

**job\_id**

- Step 1** Log in to the CCE console. In the navigation pane, choose **Clusters**. The following uses cluster management as an example to describe how to obtain the **job\_id** of the cluster that is being created.

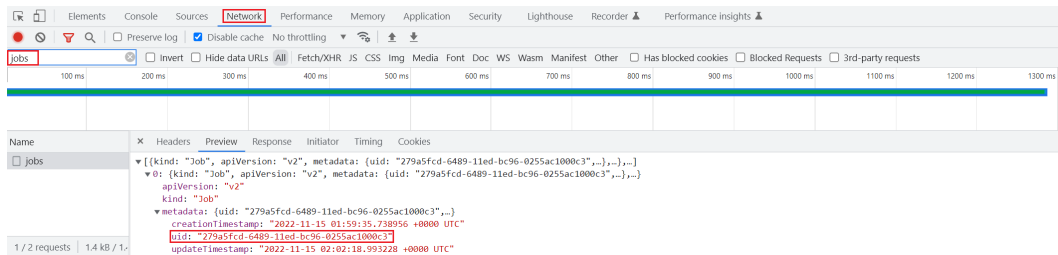
**Figure 8-5** Creating a cluster



**Step 2** Obtain the job ID.

1. If you are using Google Chrome, press **F12**. On the pane displayed on the right, click the **Network** tab.
2. Click **Operation Records** on the CCE console to view details about cluster operation records.
3. Enter **jobs** in the **Filter** text box to filter out the jobs. Select a job from the list on the left and click **Preview**. The **uid** field indicates the job UID.

**Figure 8-6** Obtaining the job ID



----End

## 8.7 Creating a VPC and Subnet

### Context

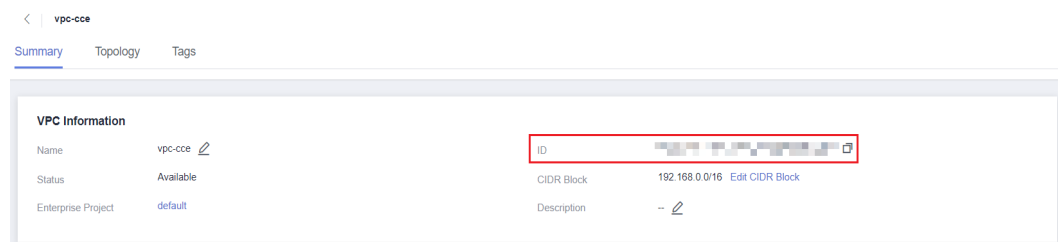
To provide a secure and isolated network environment for CCE, create a VPC before creating a cluster.

If you have already created a VPC, you do not need to create it again.

### Creating a VPC

- Step 1** On the management console, click **Service List**, and choose **Network > Virtual Private Cloud** to launch the VPC console.
- Step 2** On the VPC console, click **Create VPC** to create a VPC.
- Step 3** The created VPC is displayed in the list. Click its name and obtain the VPC ID, which will be required in **cluster creation**.

**Figure 8-7** Obtaining the VPC ID

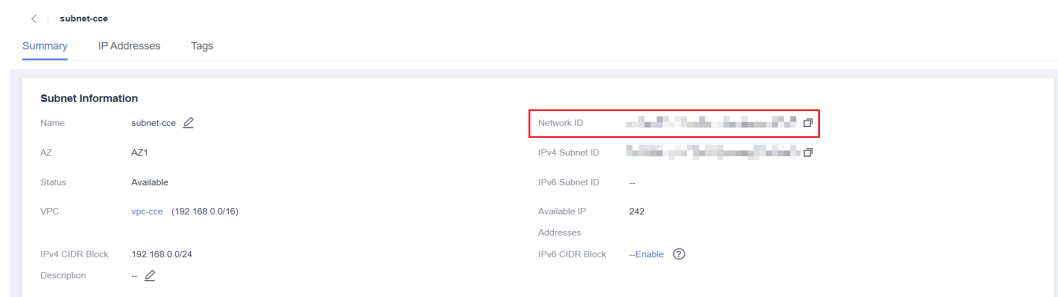


----End

## Creating a Subnet

- Step 1** On the management console, click **Service List**, and choose **Network > Virtual Private Cloud** to launch the VPC console.
- Step 2** On the **VPC console**, choose **Virtual Private Cloud > Subnets** in the navigation pane, and click **Create Subnet** in the upper right corner.
- Step 3** Create a subnet as prompted and click its name to obtain the network ID, which will be required in **cluster creation**.

**Figure 8-8** Obtaining the network ID of a subnet



----End

## 8.8 Creating a Key Pair

### Context

Create a key pair before you create a container cluster. Key pairs are used for user identity authentication upon login to a worker node.

If you have already created a key pair, you do not need to create it again.

### Procedure

- Step 1** On the management console, click **Service List**, and choose **Compute > Elastic Cloud Server** to go to the ECS console.
- Step 2** In the navigation pane, choose **Key Pair**.
- Step 3** Click **Create Key Pair** and create a key pair as prompted. For details, see **Key Pairs**.

**Step 4** After the creation is complete, a key file in the **.pem** format is generated and automatically saved to the default directory on your local computer.

----End

## 8.9 Node Flavor Description

 NOTE

Different regions support different node flavors, and node flavors may be changed or sold out. You are advised to log in to the CCE console and check whether the required node flavors are supported on the page for creating nodes.

- **CCE cluster**

CCE clusters support only 2 vCPUs and 4 GiB or higher specifications. You are advised to query node specifications on the console. For details about node specifications, see [Specifications](#).

You need to enter the specific flavor name, for example, **c6.large.2**.

The following tables list the specifications supported by CCE. (The specifications of some ECS models vary by region. The actual specifications displayed on the console are used.)

- ECS (VM)

**Table 8-13** x86 nodes

| Node Type               | Flavor                                                                         |
|-------------------------|--------------------------------------------------------------------------------|
| General computing-plus  | c7, c7n, c6s, c6, c6nl, c6x, c3, and c3ne<br>ct3<br>cx3ne<br>as7, ac7, and ac6 |
| General computing       | s7, s7n, s6, s6nl, s3, and s2<br>si3 and si2<br>sn3                            |
| General computing-basic | t6                                                                             |
| Memory-optimized        | m7, m7n, m6s, m6nl, m6, m3, and m2<br>am7 and am6                              |
| Large-memory            | e7 and e6                                                                      |
| Disk-intensive          | d7, d6, d6nl, and d2                                                           |
| Ultra-high I/O          | i7, i7n, i3, and i3nl<br>ir7, ir7n, and ir3<br>air7, air6, and ai7             |

| Node Type                  | Flavor                                                                                                            |
|----------------------------|-------------------------------------------------------------------------------------------------------------------|
| High-performance computing | h3 and hc2                                                                                                        |
| GPU-accelerated            | pi3, pi2, pi2nl, and pi1<br>p3s, p3v, p3, p3snl, p2s, p2v, p2vs, and p1<br>pc3<br>g7, g6, g6l, g5, g5r, and g5rnl |
| AI-accelerated             | ai1s, ai1, and ai2                                                                                                |

**Table 8-14** Kunpeng (Arm) nodes

| Node Type                      | Flavor         |
|--------------------------------|----------------|
| Kunpeng general computing-plus | kc1            |
| Kunpeng memory-optimized       | km1            |
| AI-accelerated                 | kai1s and kai1 |

- ECS (physical server): c6 and c7 are supported.

**Table 8-15** ECSs (PMs)

| Node Type              | Flavor                                                                                                             |
|------------------------|--------------------------------------------------------------------------------------------------------------------|
| General computing-plus | c6.22xlarge.2.physical<br>c6.22xlarge.2.physical.129nic<br>c6.22xlarge.4.physical<br>c6.22xlarge.4.physical.129nic |

- BMS:
  - x86 nodes: BMSs of the physical.d2, physical.s4, physical.c6ne, and physical.d6ne types are supported.
  - Arm nodes: BMS of the physical.a1.2xlarge type is supported.
- **CCE Turbo cluster**
  - ECS (VM)



**Table 8-16** x86 nodes

| Node Type              | Flavor                                                                               |
|------------------------|--------------------------------------------------------------------------------------|
| General computing-plus | c7, c7n, c7t, c6ne, c6sne, c6nl, and c3n<br>ac7 and as7<br>s7n and s7<br>pc5 and pc6 |
| Memory-optimized       | m7, m7n, m3n, m6ne, m6sne, and m6nl<br>am7                                           |
| Large-memory           | e7                                                                                   |
| Disk-intensive         | d7                                                                                   |
| Ultra-high I/O         | i7, i7n, and i3ne<br>ir7 and ir7n<br>air7                                            |
| GPU-accelerated        | pi3 and pi2ne<br>p3v and p2sne<br>g7 and g6ne                                        |
| AI-accelerated         | ai7 and ai2                                                                          |

**Table 8-17** Kunpeng (Arm) nodes

| Node Type                | Flavor     |
|--------------------------|------------|
| Kunpeng memory-optimized | km1n, kc1n |

- ECS (physical server): c6 and c7 are supported.

**Table 8-18** ECSs (PMs)

| Node Type              | Flavor                                                                                                             |
|------------------------|--------------------------------------------------------------------------------------------------------------------|
| General computing-plus | c6.22xlarge.2.physical<br>c6.22xlarge.2.physical.129nic<br>c6.22xlarge.4.physical<br>c6.22xlarge.4.physical.129nic |

 **NOTE**

The available specifications of IPv6 dual-stack nodes vary with regions. For details, see [Constraints](#).

## 8.10 Adding a Salt in the password Field When Creating a Node

When a node is created through the API, you need to add a salt to the **password** field to safeguard the password. The procedure is as follows:

### NOTE

The salt must be set based on the password complexity requirements:

- A string of 8–26 characters.
- Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters `!@#$%^&_+={[}];,./?`
- Cannot contain the username or the username spelled backwards.
- Cannot contain the username, the username spelled backwards, or more than two consecutive characters in the username (for Windows ECSs).

### Python

To salt a password in the Python 3.7.7 environment, perform the following steps:

### NOTE

The **python crypt** package has compatibility issues in macOS. If the package cannot be executed, run it in Linux.

1. Add `\` before `$` in the salt. Generate a ciphertext password based on the updated salt.

```
python3 -c "import crypt;print(crypt.crypt('*****', crypt.mksalt()))"
```

2. Encode the value of the **password** field using Base64.

```
echo -n '*****' | base64 | tr "\n" " " | sed s/[[:space:]]//g
```

### Java

To salt a password in the Java environment, perform the following steps:

1. Obtain a random number as the salt.

```
private static String getCharAndNumr(int length) {
 String val = "";
 Random random = new SecureRandom();
 for (int i = 0; i < length; i++) {
 // Indicates whether to output letters or digits.
 String charOrNum = random.nextInt(2) % 2 == 0 ? "char" : "num";
 // Character string
 if ("char".equalsIgnoreCase(charOrNum)) {
 // Indicates whether an upper-case or lower-case letter is obtained.
 int choice = random.nextInt(2) % 2 == 0 ? 65 : 97;
 val += (char) (choice + random.nextInt(26));
 } else if ("num".equalsIgnoreCase(charOrNum)) { // Digit
 val += String.valueOf(random.nextInt(10));
 }
 }
 return val;
}
```

2. Generate a salt.

```
private static String generateSalt() {
 String salt;
```

```

 try {
 salt = "6" + getCharAndNumr(16);
 } catch (Exception e) {
 salt = defaultSalt;
 }

 return salt;
}

```

3. Generate a ciphertext password based on the salt.

```

public static String getSaltPassword(String password) {
 if(StringUtils.isBlank(password)) {
 throw new BizException("password is empty");
 }

 String salt = generateSalt();

 Crypt crypt = new Crypt();
 return crypt.crypt(password, salt);
}

```

4. Encode the value of the **password** field using Base64.

```

(Base64.getEncoder().encodeToString(AddSaltPasswordUtil.getSaltPassword(cceNodeCreateVo.getPassword()).getBytes()))

```

5. A complete example is as follows:

```

import java.util.Base64;
import java.util.Random;
import java.security.SecureRandom;

import org.apache.commons.codec.digest.Crypt;
import org.apache.commons.lang.StringUtils;

public class PassWord {

 static String defaultSalt = null;

 public static void main(String[] args) throws Exception {
 System.out.println(Base64.getEncoder().encodeToString(PassWord.getSaltPassword("Custom
password").getBytes()));
 }

 //Generate a ciphertext password based on the salt.
 public static String getSaltPassword(String password) throws Exception {
 if(StringUtils.isBlank(password)) {
 throw new Exception("password is empty");
 }
 String salt = generateSalt();
 return Crypt.crypt(password, salt);
 }

 //Generate a salt.
 private static String generateSalt() {
 String salt;
 try {
 salt = "6" + getCharAndNumr(16);
 } catch (Exception e) {
 salt = defaultSalt;
 }
 return salt;
 }

 //Obtain a random number as the salt.
 private static String getCharAndNumr(int length) {
 String val = "";
 Random random = new SecureRandom();
 for (int i = 0; i < length; i++) {
 // Indicates whether to output letters or digits.
 String charOrNum = random.nextInt(2) % 2 == 0 ? "char" : "num";
 // Character string

```

```
 if ("char".equalsIgnoreCase(charOrNum)) {
 // Indicates whether an upper-case or lower-case letter is obtained.
 int choice = random.nextInt(2) % 2 == 0 ? 65 : 97;
 val += (char) (choice + random.nextInt(26));
 } else if ("num".equalsIgnoreCase(charOrNum)) { // Digit
 val += String.valueOf(random.nextInt(10));
 }
 }
 return val;
}
```

## Go

You can use either of the following methods to salt passwords for the Go language:

- <https://github.com/amoghe/go-crypt>
- <https://github.com/GehirnInc/crypt>

## 8.11 Maximum Number of Pods That Can Be Created on a Node

### Calculation of the Maximum Number of Pods on a Node

The maximum number of pods that can be created on a node is calculated based on the cluster type:

- For a cluster using the container tunnel network model, the value depends only on [the maximum number of pods on a node](#).
- For clusters using the VPC network model, the value depends on [the maximum number of pods on a node](#) and [the minimum number of container IP addresses that can be allocated to a node](#). It is recommended that the maximum number of pods on a node be less than or equal to the number of container IP addresses that can be allocated to the node. Otherwise, pods may fail to be scheduled.
- For CCE Turbo clusters using the Cloud Native Network 2.0 model, the value depends on [the maximum number of pods on a node](#) and [the minimum number of ENIs on a CCE Turbo cluster node](#). It is recommended that the maximum number of pods on a node be less than or equal to the number of ENIs on the node. Otherwise, pods may fail to be scheduled.

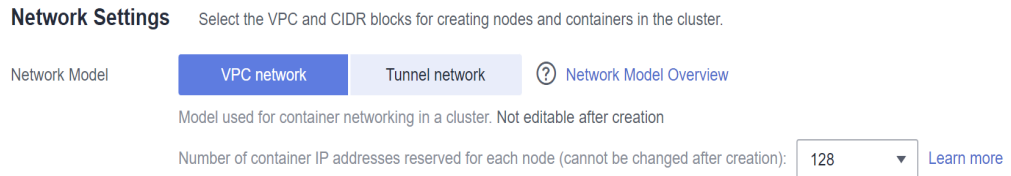
### Number of Container IP Addresses That Can Be Allocated on a Node

If you select **VPC network** for **Network Model** when creating a CCE cluster, you also need to set the number of container IP addresses that can be allocated to each node (`alpha.cce/fixPoolMask`). If the pod uses the host network (**hostNetwork: true**), the pod does not occupy the IP address of the allocatable container network. For details, see [Container Network vs. Host Network](#).

This parameter affects the maximum number of pods that can be created on a node. Each pod occupies an IP address (when the [container network](#) is used). If the number of available IP addresses is insufficient, pods cannot be created. If the

pod uses the host network (**hostNetwork: true**), the pod does not occupy the IP address of the allocatable container network.

**Figure 8-9** Specifying the number of allocatable container IP addresses on a node in the VPC network model



By default, a node occupies three container IP addresses (network address, gateway address, and broadcast address). Therefore, the number of container IP addresses that can be allocated to a node equals the number of selected container IP addresses minus 3. For example, in the preceding figure, the number of container IP addresses that can be allocated to a node is 125 (128 - 3).

## Maximum Number of Pods on a Node

When creating a node, you can configure the maximum number of pods (maxPods) that can be created on the node. This parameter is a configuration item of kubelet and determines the maximum number of pods that can be created by kubelet.

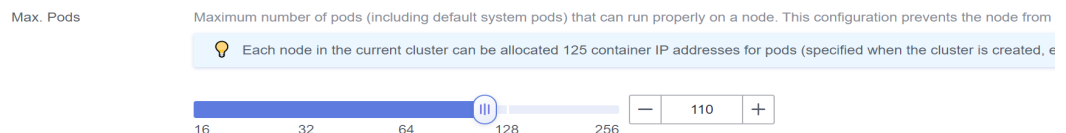
### NOTICE

For nodes in the default node pool (**DefaultPool**), the maximum number of pods cannot be changed after the nodes are created.

After a node in a custom node pool is created, you can modify the **max-pods** parameter in the node pool configuration to change the maximum number of pods on the node. For details, see [Configuring a Node Pool](#).

By default, the maximum number of pods on a node can be adjusted to 256. To increase the deployment density on a node, submit a [service ticket](#) to increase the maximum number of pods on a node, which can be 512.

**Figure 8-10** Specifying the maximum number of pods on a node



**Table 8-19** lists the default maximum number of pods on a node based on node specifications.

**Table 8-19** Default maximum number of pods on a node

| Memory         | Max. Pods |
|----------------|-----------|
| 4 GB           | 20        |
| 8 GB           | 40        |
| 16 GB          | 60        |
| 32 GB          | 80        |
| 64 GB or above | 110       |

## Number of Node ENIs (CCE Turbo Clusters)

In a CCE Turbo cluster, ECS nodes use sub-ENIs and BMS nodes use ENIs. The maximum number of pods that can be created on a node depends on the number of ENIs that can be used by the node.

**Figure 8-11** Node ENIs

| Flavor                                      | vCPUs   Memory | Assured/Maximum Bandwidth | Packets Per Second (PPS) | Max. Pods |
|---------------------------------------------|----------------|---------------------------|--------------------------|-----------|
| <input checked="" type="radio"/> c7.large.2 | 2cores   4GB   | 0.84.0 Gbit/s             | 400,000 pps              | 16        |
| <input type="radio"/> c7.large.4            | 2cores   8GB   | 0.84.0 Gbit/s             | 400,000 pps              | 16        |
| <input type="radio"/> c7.xlarge.2           | 4cores   8GB   | 1.68.0 Gbit/s             | 800,000 pps              | 32        |

## Container Network vs. Host Network

When creating a pod, you can select the container network or host network for the pod.

- Container network (default): **Each pod is assigned an IP address by the cluster networking add-ons, which occupies the IP addresses of the container network.**
- Host network: The pod uses the host network (**hostNetwork: true** needs to be configured for the pod) and occupies the host port. The pod IP address is the host IP address. The pod does not occupy the IP addresses of the container network. To use the host network, you must confirm whether the container ports conflict with the host ports. Do not use the host network unless you know exactly which host port is used by which container.

## 8.12 Node OS

### Mappings Between Cluster Versions and OS Versions

The following table lists the mappings between released cluster versions and OS versions.

**Table 8-20** ECS (VM) OS

| OS                             | Cluster Version | CCE Standard Cluster |                                    | CCE Turbo Cluster        | Latest Kernel                           |
|--------------------------------|-----------------|----------------------|------------------------------------|--------------------------|-----------------------------------------|
|                                |                 | VPC Network          | Tunnel Network                     | Cloud Native 2.0 Network |                                         |
| Huawei Cloud EulerOS 2.0       | v1.29           | √                    | √                                  | √                        | 5.10.0-60.18.0.50.r1083_58.hce2.x86_64  |
|                                | v1.28           | √                    | √                                  | √                        | 5.10.0-60.18.0.50.r1083_58.hce2.x86_64  |
|                                | v1.27           | √                    | Supported in v1.27.3-r0 or later.  | √                        | 5.10.0-60.18.0.50.r1083_58.hce2.x86_64  |
|                                | v1.25           | √                    | Supported in v1.25.6-r0 or later.  | √                        | 5.10.0-60.18.0.50.r1083_58.hce2.x86_64  |
|                                | v1.23           | √                    | Supported in v1.23.11-r0 or later. | √                        | 5.10.0-60.18.0.50.r1083_58.hce2.x86_64  |
| Huawei Cloud EulerOS 2.0 (Arm) | v1.29           | √                    | √                                  | √                        | 5.10.0-60.18.0.50.r1083_58.hce2.aarch64 |
|                                | v1.28           | √                    | √                                  | √                        | 5.10.0-60.18.0.50.r1083_58.hce2.aarch64 |
|                                | v1.27           | √                    | Supported in v1.27.3-r0 or later.  | √                        | 5.10.0-60.18.0.50.r1083_58.hce2.aarch64 |
|                                | v1.25           | √                    | Supported in v1.25.6-r0 or later.  | √                        | 5.10.0-60.18.0.50.r1083_58.hce2.aarch64 |

| OS                       | Cluster Version | CCE Standard Cluster |                                    | CCE Turbo Cluster        | Latest Kernel                           |
|--------------------------|-----------------|----------------------|------------------------------------|--------------------------|-----------------------------------------|
|                          |                 | VPC Network          | Tunnel Network                     | Cloud Native 2.0 Network |                                         |
|                          | v1.23           | √                    | Supported in v1.23.11-r0 or later. | √                        | 5.10.0-60.18.0.50.r1083_58.hce2.aarch64 |
| Ubuntu 22.04             | v1.29           | √                    | x                                  | √                        | 5.15.0-92-generic                       |
|                          | v1.28           | √                    | x                                  | √                        | 5.15.0-92-generic                       |
|                          | v1.27           | √                    | x                                  | √                        | 5.15.0-92-generic                       |
|                          | v1.25           | √                    | x                                  | √                        | 5.15.0-92-generic                       |
|                          | v1.23           | √                    | x                                  | √                        | 5.15.0-92-generic                       |
| Huawei Cloud EulerOS 1.1 | v1.29           | √                    | √                                  | √                        | 3.10.0-1160.76.2.hce1c.x86_64           |
|                          | v1.28           | √                    | √                                  | √                        | 3.10.0-1160.76.2.hce1c.x86_64           |
|                          | v1.27           | √                    | √                                  | √                        | 3.10.0-1160.76.2.hce1c.x86_64           |
|                          | v1.25           | √                    | √                                  | √                        | 3.10.0-1160.76.2.hce1c.x86_64           |
|                          | v1.23           | √                    | √                                  | √                        | 3.10.0-1160.76.2.hce1c.x86_64           |
|                          | v1.21           | √                    | √                                  | √                        | 3.10.0-1160.76.2.hce1c.x86_64           |
| CentOS Linux release 7.6 | v1.29           | √                    | √                                  | √                        | 3.10.0-1160.108.1.el7.x86_64            |
|                          | v1.28           | √                    | √                                  | √                        | 3.10.0-1160.108.1.el7.x86_64            |
|                          | v1.27           | √                    | √                                  | √                        | 3.10.0-1160.108.1.el7.x86_64            |
|                          | v1.25           | √                    | √                                  | √                        | 3.10.0-1160.108.1.el7.x86_64            |
|                          | v1.23           | √                    | √                                  | √                        | 3.10.0-1160.108.1.el7.x86_64            |



| OS    | Cluster Version                  | CCE Standard Cluster |                | CCE Turbo Cluster        | Latest Kernel                             |
|-------|----------------------------------|----------------------|----------------|--------------------------|-------------------------------------------|
|       |                                  | VPC Network          | Tunnel Network | Cloud Native 2.0 Network |                                           |
|       | v1.21                            | √                    | √              | √                        | 3.10.0-1160.108.1.el7.x86_64              |
|       | v1.19 (End of maintenance)       | √                    | √              | √                        | 3.10.0-1160.108.1.el7.x86_64              |
|       | v1.17.17 (end of maintenance)    | √                    | √              | √                        | 3.10.0-1160.15.2.el7.x86_64               |
|       | v1.17.9 (end of maintenance)     | √                    | √              | √                        | 3.10.0-1062.12.1.el7.x86_64               |
|       | v1.15.11 (end of maintenance)    | √                    | √              | √                        | 3.10.0-1062.12.1.el7.x86_64               |
|       | v1.15.6-r1 (end of maintenance)  | √                    | √              | √                        | 3.10.0-1062.1.1.el7.x86_64                |
|       | v1.13.10-r1 (end of maintenance) | √                    | √              | √                        | 3.10.0-957.21.3.el7.x86_64                |
|       | v1.13.7-r0 (end of maintenance)  | √                    | √              | √                        | 3.10.0-957.21.3.el7.x86_64                |
|       | EulerOS release 2.9              | v1.29                | √              | √                        | √                                         |
| v1.28 |                                  | √                    | √              | √                        | 4.18.0-147.5.1.6.h1152.eulerosv2r9.x86_64 |

| OS                        | Cluster Version            | CCE Standard Cluster |                | CCE Turbo Cluster        | Latest Kernel                                   |
|---------------------------|----------------------------|----------------------|----------------|--------------------------|-------------------------------------------------|
|                           |                            | VPC Network          | Tunnel Network | Cloud Native 2.0 Network |                                                 |
|                           | v1.27                      | √                    | √              | √                        | 4.18.0-147.5.1.6.h1152.eulerosv2r9.x86_64       |
|                           | v1.25                      | √                    | √              | √                        | 4.18.0-147.5.1.6.h1152.eulerosv2r9.x86_64       |
|                           | v1.23                      | √                    | √              | √                        | 4.18.0-147.5.1.6.h1152.eulerosv2r9.x86_64       |
|                           | v1.21                      | √                    | √              | √                        | 4.18.0-147.5.1.6.h1152.eulerosv2r9.x86_64       |
|                           | v1.19 (End of maintenance) | √                    | √              | √                        | 4.18.0-147.5.1.6.h1152.eulerosv2r9.x86_64       |
| EulerOS release 2.9 (Arm) | v1.29                      | √                    | √              | √                        | 4.19.90-vhulk2103.1.0.h1144.eulerosv2r9.aarch64 |
|                           | v1.28                      | √                    | √              | √                        | 4.19.90-vhulk2103.1.0.h1144.eulerosv2r9.aarch64 |
|                           | v1.27                      | √                    | √              | √                        | 4.19.90-vhulk2103.1.0.h1144.eulerosv2r9.aarch64 |
|                           | v1.25                      | √                    | √              | √                        | 4.19.90-vhulk2103.1.0.h1144.eulerosv2r9.aarch64 |
|                           | v1.23                      | √                    | √              | √                        | 4.19.90-vhulk2103.1.0.h1144.eulerosv2r9.aarch64 |

| OS                                            | Cluster Version               | CCE Standard Cluster |                | CCE Turbo Cluster        | Latest Kernel                                   |
|-----------------------------------------------|-------------------------------|----------------------|----------------|--------------------------|-------------------------------------------------|
|                                               |                               | VPC Network          | Tunnel Network | Cloud Native 2.0 Network |                                                 |
|                                               | v1.21                         | √                    | √              | √                        | 4.19.90-vhulk2103.1.0.h1144.eulerosv2r9.aarch64 |
|                                               | v1.19 (End of maintenance)    | √                    | √              | √                        | 4.19.90-vhulk2103.1.0.h1144.eulerosv2r9.aarch64 |
| EulerOS release 2.8 (Arm, end of maintenance) | v1.27 or later                | x                    | x              | x                        | None                                            |
|                                               | 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  |

| OS                                       | Cluster Version               | CCE Standard Cluster |                | CCE Turbo Cluster        | Latest Kernel                                  |
|------------------------------------------|-------------------------------|----------------------|----------------|--------------------------|------------------------------------------------|
|                                          |                               | VPC Network          | Tunnel Network | Cloud Native 2.0 Network |                                                |
|                                          | v1.15.11 (end of maintenance) | √                    | √              | √                        | 4.19.36-vhulk1907.1.0.h702.eulerosv2r8.aarch64 |
| EulerOS release 2.5 (end of maintenance) | v1.27 or later                | x                    | x              | x                        | None                                           |
|                                          | v1.25                         | √                    | √              | √                        | 3.10.0-862.14.1.5.h687.eulerosv2r7.x86_64      |
|                                          | 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 (end of maintenance) | √                    | √              | √                        | 3.10.0-862.14.1.5.h470.eulerosv2r7.x86_64      |
|                                          | v1.17.9 (end of maintenance)  | √                    | √              | √                        | 3.10.0-862.14.1.5.h428.eulerosv2r7.x86_64      |
|                                          | v1.15.11 (end of maintenance) | √                    | √              | √                        | 3.10.0-862.14.1.5.h428.eulerosv2r7.x86_64      |

| OS                                              | Cluster Version                  | CCE Standard Cluster |                | CCE Turbo Cluster        | Latest Kernel                             |
|-------------------------------------------------|----------------------------------|----------------------|----------------|--------------------------|-------------------------------------------|
|                                                 |                                  | VPC Network          | Tunnel Network | Cloud Native 2.0 Network |                                           |
|                                                 | v1.15.6-r1 (end of maintenance)  | √                    | √              | √                        | 3.10.0-862.14.1.5.h328.eulerosv2r7.x86_64 |
|                                                 | v1.13.10-r1 (end of maintenance) | √                    | √              | √                        | 3.10.0-862.14.1.2.h249.eulerosv2r7.x86_64 |
|                                                 | v1.13.7-r0 (end of maintenance)  | √                    | √              | √                        | 3.10.0-862.14.1.0.h197.eulerosv2r7.x86_64 |
| Ubuntu 18.04 server 64-bit (end of maintenance) | v1.27 or later                   | x                    | x              | x                        | None                                      |
|                                                 | v1.25                            | √                    | x              | √                        | 4.15.0-171-generic                        |
|                                                 | v1.23                            | √                    | x              | √                        | 4.15.0-171-generic                        |
|                                                 | v1.21                            | √                    | x              | √                        | 4.15.0-171-generic                        |
|                                                 | v1.19.16                         | √                    | x              | √                        | 4.15.0-171-generic                        |
|                                                 | v1.19.8                          | √                    | x              | √                        | 4.15.0-136-generic                        |
|                                                 | v1.17.17                         | √                    | x              | √                        | 4.15.0-136-generic                        |

**Table 8-21** ECS (PM) OS

| OS                   | Cluster Version               | CCE Standard Cluster |                | CCE Turbo Cluster        | Latest Kernel                               |
|----------------------|-------------------------------|----------------------|----------------|--------------------------|---------------------------------------------|
|                      |                               | VPC Network          | Tunnel Network | Cloud Native 2.0 Network |                                             |
| EulerOS release 2.10 | v1.29                         | √                    | √              | √                        | 4.18.0-147.5.2.15.h1109.eulerosv2r10.x86_64 |
|                      | v1.28                         | √                    | √              | √                        | 4.18.0-147.5.2.15.h1109.eulerosv2r10.x86_64 |
|                      | v1.27                         | √                    | √              | √                        | 4.18.0-147.5.2.15.h1109.eulerosv2r10.x86_64 |
|                      | v1.25                         | √                    | √              | √                        | 4.18.0-147.5.2.15.h1109.eulerosv2r10.x86_64 |
|                      | v1.23                         | √                    | √              | √                        | 4.18.0-147.5.2.15.h1109.eulerosv2r10.x86_64 |
|                      | v1.21                         | √                    | √              | √                        | 4.18.0-147.5.2.15.h1109.eulerosv2r10.x86_64 |
|                      | v1.19.16 (End of maintenance) | √                    | √              | √                        | 4.18.0-147.5.2.15.h1109.eulerosv2r10.x86_64 |

**Table 8-22** BMS OS

| OS                                                                             | Cluster Version | CCE Standard Cluster |                | CCE Turbo Cluster        | Latest Kernel                            |
|--------------------------------------------------------------------------------|-----------------|----------------------|----------------|--------------------------|------------------------------------------|
|                                                                                |                 | VPC Network          | Tunnel Network | Cloud Native 2.0 Network |                                          |
| EulerOS release 2.9 (restricted use. Submit a service ticket to apply for it.) | v1.28           | √                    | √              | x                        | 4.18.0-147.5.1.6.h841.eulerosv2r9.x86_64 |
|                                                                                | v1.27           | √                    | √              | x                        | 4.18.0-147.5.1.6.h841.eulerosv2r9.x86_64 |
|                                                                                | v1.25           | √                    | √              | x                        | 4.18.0-147.5.1.6.h841.eulerosv2r9.x86_64 |
|                                                                                | v1.23           | √                    | √              | x                        | 4.18.0-147.5.1.6.h841.eulerosv2r9.x86_64 |
|                                                                                | v1.21           | √                    | √              | x                        | 4.18.0-147.5.1.6.h841.eulerosv2r9.x86_64 |
|                                                                                | v1.19           | √                    | √              | x                        | 4.18.0-147.5.1.6.h841.eulerosv2r9.x86_64 |
| EulerOS release 2.3 (end of maintenance)                                       | v1.27 or later  | x                    | x              | x                        | None                                     |
|                                                                                | v1.25           | √                    | √              | x                        | 3.10.0-514.41.4.28.h62.x86_64            |
|                                                                                | v1.23           | √                    | √              | x                        | 3.10.0-514.41.4.28.h62.x86_64            |
|                                                                                | v1.21           | √                    | √              | x                        | 3.10.0-514.41.4.28.h62.x86_64            |
|                                                                                | v1.19           | √                    | √              | x                        | 3.10.0-514.41.4.28.h62.x86_64            |
|                                                                                | v1.17           | √                    | √              | x                        | 3.10.0-514.41.4.28.h62.x86_64            |
|                                                                                | v1.15.11        | √                    | √              | x                        | 3.10.0-514.41.4.28.h62.x86_64            |

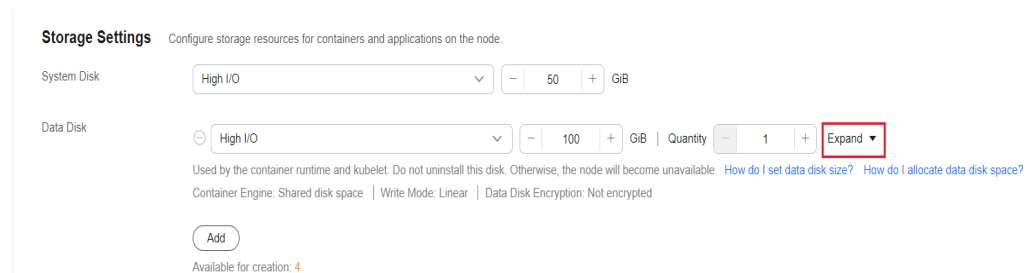
## 8.13 Data Disk Space Allocation

This section describes how to allocate data disk space to nodes so that you can configure the data disk space accordingly.

### Allocating Data Disk Space

When creating a node, configure data disks for the node. You can also click **Expand** and customize the data disk space allocation for the node.

**Figure 8-12** Allocating data disk space



- **Space Allocation for Container Engines**
  - Specified disk space: CCE divides the data disk space for two parts by default. One part is used to store 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.
    - Container engine and container image space (90% by default): stores the container runtime working directories, container image data, and image metadata.
    - kubelet and emptyDir space (10% by default): stores pod configuration files, secrets, and mounted storage such as emptyDir volumes.
  - Shared disk space: In clusters of v1.21.10-r0, v1.23.8-r0, v1.25.3-r0, or later versions, CCE allows **a container engine (Docker/containerd) and kubelet components to share data disk space.**
- **Space Allocation for Pods:** indicates the basesize of a pod. You can set an upper limit for the disk space occupied by each workload pod (including the space occupied by container images). This setting prevents the pods from taking all the disk space available, which may cause service exceptions. It is recommended that the value is less than or equal to 80% of the container engine space. This parameter is related to the node OS and container storage roots and is not supported in some scenarios. For details, see **Mapping Between OS and Container Storage Rootfs.**
- Write Mode
  - **Linear:** A linear logical volume integrates one or more physical volumes. Data is written to the next physical volume when the previous one is used up.



- **Striped:** available only if there are at least two data disks. A striped logical volume stripes data into blocks of the same size and stores them in multiple physical volumes in sequence. This allows data to be concurrently read and written. A storage pool consisting of striped volumes cannot be scaled-out.

## Space Allocation for Container Engines

For nodes using a data disk shared between a container engine and kubelet components, the container storage Rootfs is of the **OverlayFS** type. For details about data disk space allocation, see [Data Disk Shared Between a Container Engine and kubelet Components](#).

For a node using a non-shared data disk (100 GiB for example), the division of the disk space varies depending on the container storage Rootfs type **Device Mapper** or **OverlayFS**. For details about the container storage Rootfs corresponding to different OSs, see [Mapping Between OS and Container Storage Rootfs](#).

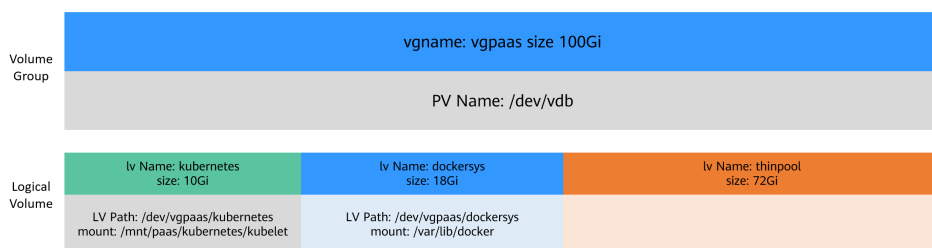
- **Rootfs (Device Mapper)**

By default, the container engine and image space, occupying 90% of the data disk, can be divided into the following two parts:

- The **/var/lib/docker** directory is used as the Docker working directory and occupies 20% of the container engine and container image space by default. (Space size of the **/var/lib/docker** directory = **Data disk space x 90% x 20%**)
- The thin pool is used to store container image data, image metadata, and container data, and occupies 80% of the container engine and container image space by default. (Thin pool space = **Data disk space x 90% x 80%**)

The thin pool is dynamically mounted. You can view it by running the **lsblk** command on a node, but not the **df -h** command.

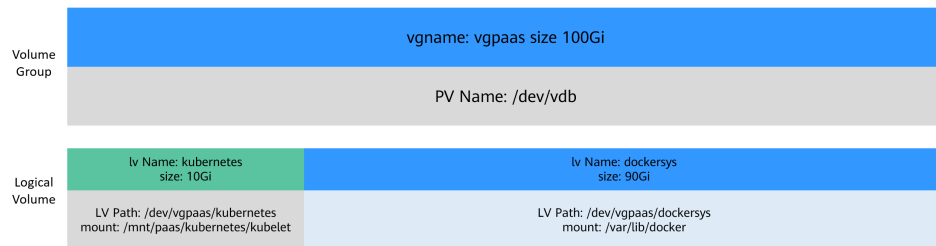
**Figure 8-13** Space allocation for container engines of Device Mapper



- **Rootfs (OverlayFS)**

No separate thin pool. The entire container engine and container image space (90% of the data disk by default) are in the **/var/lib/docker** directory.

**Figure 8-14** Space allocation for container engines of OverlayFS



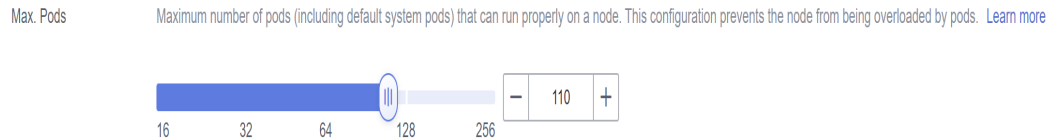
## Space Allocation for Pods

The customized pod container space (basesize) is related to the node OS and container storage Rootfs. For details about the container storage Rootfs, see [Mapping Between OS and Container Storage Rootfs](#).

- Device Mapper supports custom pod basesize. The default value is 10 GiB.
- In OverlayFS mode, the pod container space is not limited by default.

When configuring **basesize**, consider the maximum number of pods on a node. The container engine space should be greater than the total disk space used by containers. Formula: **the container engine space and container image space (90% by default) > Number of containers x basesize**. Otherwise, the container engine space allocated to the node may be insufficient and the container cannot be started.

**Figure 8-15** Maximum number of pods



For nodes that support **basesize**, when Device Mapper is used, although you can limit the size of the **/home** directory of a single container (to 10 GB by default), all containers on the node still share the thin pool of the node for storage. They are not completely isolated. When the sum of the thin pool space used by certain containers reaches the upper limit, other containers cannot run properly.

In addition, after a file is deleted in the **/home** directory of the container, the thin pool space occupied by the file is not released immediately. Therefore, even if **basesize** is set to 10 GB, the thin pool space occupied by files keeps increasing until 10 GB when files are created in the container. The space released after file deletion will be reused but after a while. If **the number of containers on the node multiplied by basesize** is greater than the thin pool space size of the node, there is a possibility that the thin pool space has been used up.

## Mapping Between OS and Container Storage Rootfs

**Table 8-23** Node OSs and container engines in CCE clusters

| OS                       | Container Storage Rootfs                                                                                   | Customized Basesize                                                                                                                                                                                                                                                                                       |
|--------------------------|------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CentOS 7.x               | Clusters of v1.19.16 and earlier use Device Mapper.<br>Clusters of v1.19.16 and later use OverlayFS.       | Supported when Rootfs is set to Device Mapper and the container engine is Docker. The default value is 10 GiB.<br>Not supported when Rootfs is set to OverlayFS.                                                                                                                                          |
| EulerOS 2.3              | Device Mapper                                                                                              | Supported only when the container engine is Docker. The default value is 10 GiB.                                                                                                                                                                                                                          |
| EulerOS 2.5              | Device Mapper                                                                                              | Supported only when the container engine is Docker. The default value is 10 GiB.                                                                                                                                                                                                                          |
| EulerOS 2.8              | Clusters of v1.19.16-r2 and earlier use Device Mapper.<br>Clusters of v1.19.16-r2 and later use OverlayFS. | Supported when Rootfs is set to Device Mapper and the container engine is Docker. The default value is 10 GiB.<br>Supported only when Rootfs is set to OverlayFS and the container engine is Docker. There are no limits by default.                                                                      |
| EulerOS 2.9              | OverlayFS                                                                                                  | Supported only by clusters of v1.19.16, v1.21.3, v1.23.3, or later. There are no limits by default.<br>Not supported if the cluster versions are earlier than v1.19.16, v1.21.3, or v1.23.3.                                                                                                              |
| EulerOS 2.10             | OverlayFS                                                                                                  | Supported only by Docker clusters of versions earlier than v1.23.14-r0, v1.25.9-r0, v1.27.6-r0, or v1.28.4-r0. There are no limits by default.<br>Supported by both Docker and containerd clusters of v1.23.14-r0, v1.25.9-r0, v1.27.6-r0, v1.28.4-r0, or later versions. There are no limits by default. |
| Ubuntu 18.04             | OverlayFS                                                                                                  | Not supported.                                                                                                                                                                                                                                                                                            |
| Huawei Cloud EulerOS 1.1 | OverlayFS                                                                                                  | Not supported.                                                                                                                                                                                                                                                                                            |

| OS                       | Container Storage Rootfs | Customized Basesize                                                                                                                                                                                                                                                                                           |
|--------------------------|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Huawei Cloud EulerOS 2.0 | OverlayFS                | Supported only by Docker clusters of versions earlier than v1.23.14-r0, v1.25.9-r0, v1.27.6-r0, or v1.28.4-r0. There are no limits by default.<br><br>Supported by both Docker and containerd clusters of v1.23.14-r0, v1.25.9-r0, v1.27.6-r0, v1.28.4-r0, or later versions. There are no limits by default. |

**Table 8-24** Node OSs and container engines in CCE Turbo clusters

| OS                       | Container Storage Rootfs                             | Customized Basesize                                                                                                                                                                                                                                                                                                                                              |
|--------------------------|------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CentOS 7.x               | OverlayFS                                            | Not supported.                                                                                                                                                                                                                                                                                                                                                   |
| Ubuntu 18.04             | OverlayFS                                            | Not supported.                                                                                                                                                                                                                                                                                                                                                   |
| EulerOS 2.9              | ECS VMs use OverlayFS.<br>ECS PMs use Device Mapper. | Supported only when Rootfs is set to OverlayFS and the container engine is Docker. There are no limits by default. Supported by both Docker and containerd clusters of v1.23.14-r0, v1.25.9-r0, v1.27.6-r0, v1.28.4-r0, or later versions.<br><br>Supported when Rootfs is set to Device Mapper and the container engine is Docker. The default value is 10 GiB. |
| Huawei Cloud EulerOS 1.1 | OverlayFS                                            | Not supported.                                                                                                                                                                                                                                                                                                                                                   |
| Huawei Cloud EulerOS 2.0 | OverlayFS                                            | Supported only by Docker clusters of versions earlier than v1.23.14-r0, v1.25.9-r0, v1.27.6-r0, or v1.28.4-r0. There are no limits by default.<br><br>Supported by both Docker and containerd clusters of v1.23.14-r0, v1.25.9-r0, v1.27.6-r0, v1.28.4-r0, or later versions. There are no limits by default.                                                    |

## Garbage Collection Policies for Container Images

When the container engine space is insufficient, image garbage collection is triggered.

The policy for garbage collecting images takes two factors into consideration: **HighThresholdPercent** and **LowThresholdPercent**. Disk usage exceeding the high threshold (default: 80%) will trigger garbage collection. The garbage collection will delete least recently used images until the low threshold (default: 70%) is met.

## Recommended Configuration for the Container Engine Space

- The container engine space should be greater than the total disk space used by containers. Formula: **Container engine space > Number of containers x basesize**
- You are advised to create and delete files of containerized services in local storage volumes (such as emptyDir and hostPath volumes) or cloud storage directories mounted to the containers. In this way, the thin pool space is not occupied. emptyDir volumes occupy the kubelet space. Therefore, properly plan the size of the kubelet space.
- You can deploy services on nodes that use the OverlayFS (for details, see [Mapping Between OS and Container Storage Rootfs](#)) so that the disk space occupied by files created or deleted in containers can be released immediately.

## Data Disk Shared Between a Container Engine and kubelet Components

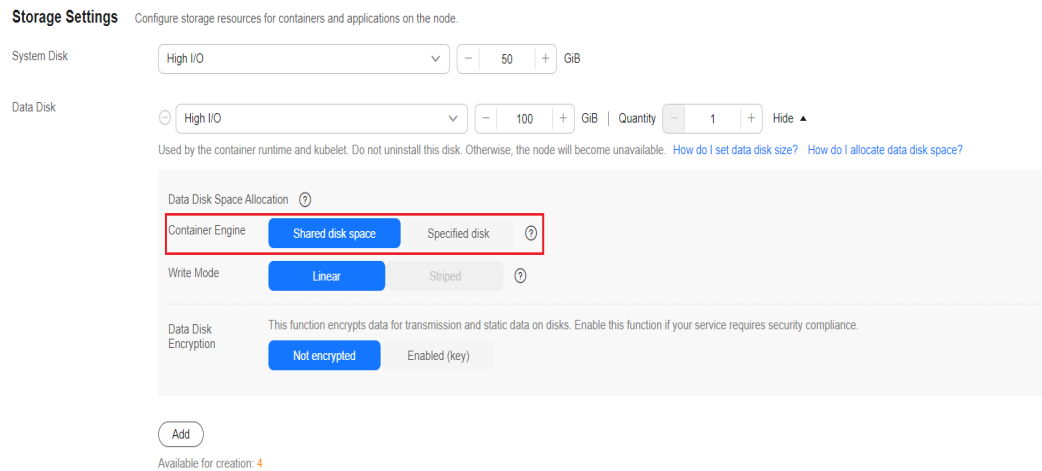
Docker/containerd and kubelet components share the space of a data disk.

---

### NOTICE

- This function is available only to clusters of v1.21.10-r0, v1.23.8-r0, v1.25.3-r0, or later versions.
  - If Rootfs is set to OverlayFS, shared data disks are supported. If Rootfs is set to Device Mapper, shared data disks are not supported.
  - If you have installed an NPD add-on in the cluster, upgrade the add-on to v1.18.10 or later. Otherwise, false alarms will be generated.
  - If you have installed a log-agent add-on in the cluster, upgrade the add-on to v1.3.0 or later. Otherwise, log collection will be affected.
  - If you have installed ICAgent in the cluster, upgrade it to v5.12.140 or later. Otherwise, log collection will be affected. For details about how to view or upgrade an ICAgent version, see [CCE Access](#).
-

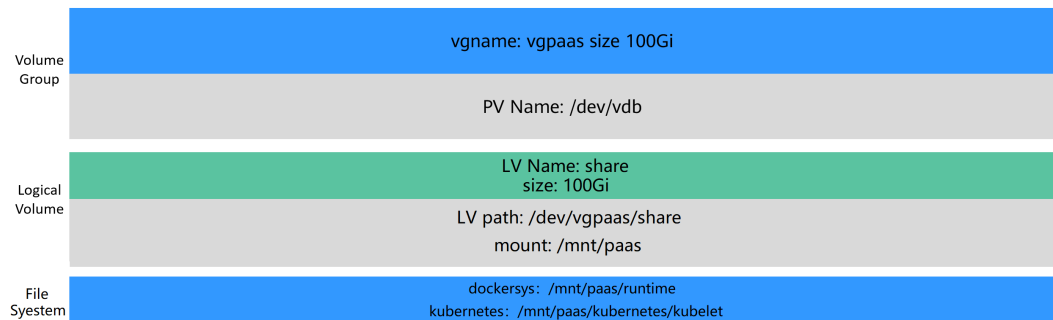
**Figure 8-16** Configuration for sharing disk space



For nodes using a shared data disk, the container storage Rootfs is of the **OverlayFS** type. After such a node is created, the data disk space (for example, 100 GiB) will not be divided for the container engines, container images, and kubelet components. The data disk is mounted to **/mnt/paas**, and the storage space is divided using two file systems.

- dockersys: /mnt/paas/runtime
- Kubernetes: /mnt/paas/kubernetes/kubelet

**Figure 8-17** Allocating the storage space of a shared data disk



## Common Issues

[How Do I Expand the Storage Capacity of a Container?](#)

[Expanding the Disk Capacity of a Node in a CCE Cluster](#)

## 8.14 Attaching Disks to a Node

### Challenges

In disk planning and striped logical disk creation, it is difficult for users to flexibly attach and partition disks when creating a node.

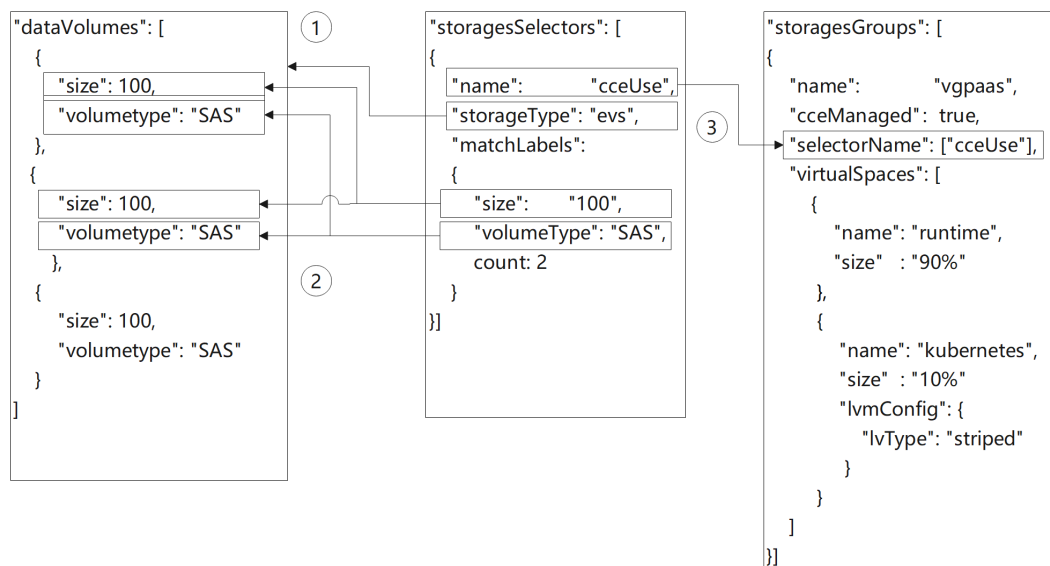
During node creation, the **storage** field selects a data disk based on parameters, such as the disk size and disk type, to prevent failures in node creation, resetting, migration, and management caused by drive letter matching failures. For example, if **DockerLVMConfigOverride.diskType** in the NodeExtendParam field is set to **evs** during node creation, C7 nodes will not be created.

## Solution

This section provides details about the **storage** field used in node creation so that you can implement complex disk selection and partitioning by calling the node creation API.

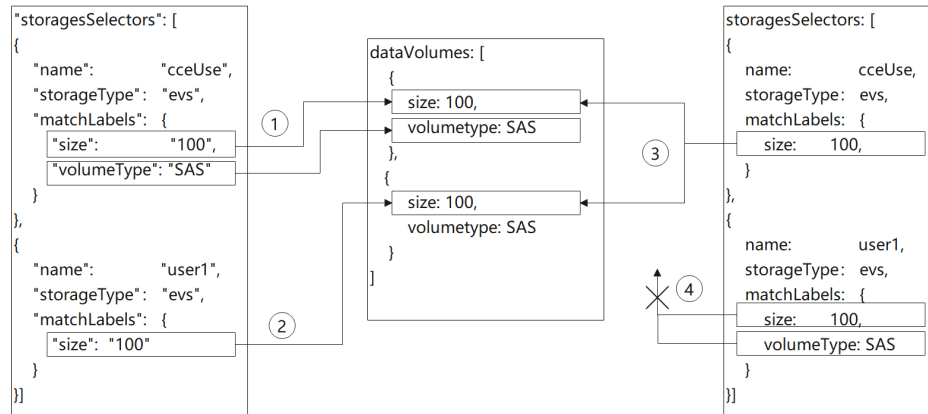
The **storage** field consists of **storageSelectors** and **storageGroups**. The **storageSelectors** field is responsible for disk selection, and the **storageGroups** field is responsible for disk processing.

The basic logic for field matching is as follows:



1. **storageSelectors** selects an EVS disk or a local disk based on the value of **storageType**.
  - a. Local disks do not support exact match. All local disks will be selected as data disks.  
If some local disks need to be reserved, occupy the disks in the pre-installation script. The script is similar to the following:
 

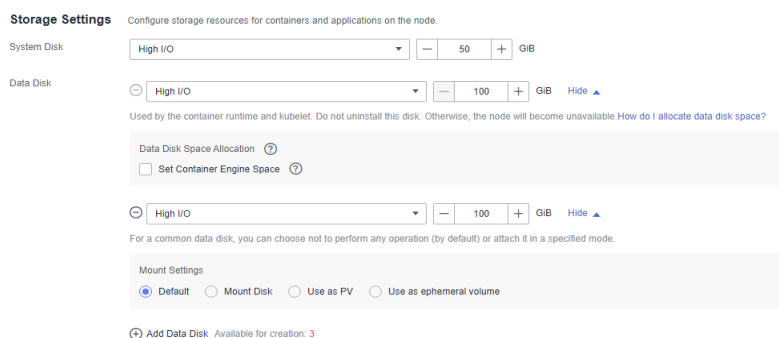
```
prepare
vgName=vg-test
storageDevice=/dev/vdb
vgcreate
vgcreate ${vgName} ${storageDevice}
```
  - b. EVS disks match the disks created in **dataVolumes** based on the settings of **matchLabels**.
2. Policies have priorities to match **matchLabels**. The **matchLabels** policy nearest to **storageSelectors** has a higher priority and the disk nearest to **dataVolumes** will be preferentially matches. As **matchLabels** uses loose matching, you are advised to place the **matchLabels** policy with a small matching range on the top. For example:



- a. In step 1, the first disk in **dataVolumes** is matched by the EVS disk whose size is 100 GiB and storage class is SAS. In step 2, the second disk in **dataVolumes** is matched by the EVS disk whose size is 100 GiB because the first disk has been selected.
- b. In step 3, two disks in **dataVolumes** can be matched because **volumeType** or **count** is not specified in **matchLabels**. In this case, no disk is available for matching in step 4.
3. **storageGroups** associates with **storageSelectors** based on **selectorName**. Finally, two 100 GiB disks are selected. The CCE backend groups the two PVs into a volume group (VG) and divides the VG into two logical volumes (LVs) in the ratio of 9:1. 10% of Kubernetes LVs are partitioned in striped mode. 90% runtime LVs are partitioned in linear mode by default because **runtimeConfig** is not configured.

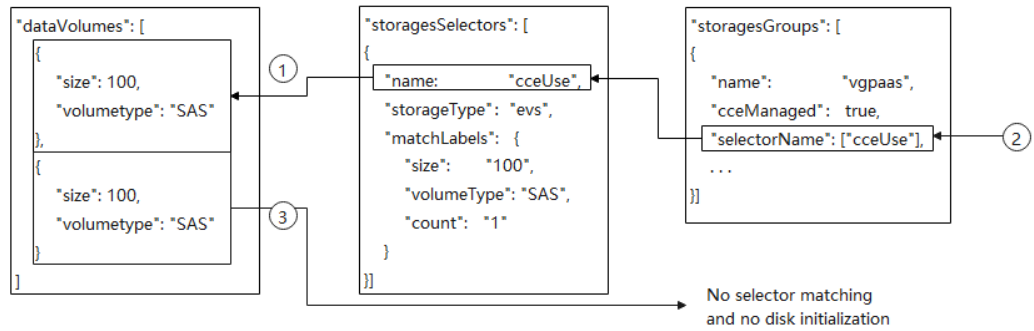
## Creating a Raw Disk

During node creation on the CCE console, click **Add Data Disk**. Then, click **Expand** next to the newly added data disk and select **Default**. The created disk is a raw disk.



The following figure shows the API calling logic.





1. The **cceUse** selector matches a 100 GiB data disk.
2. The selected disk is managed by CCE and used as a data disk.
3. The other 100 GiB data disk created in **dataVolumes** is not selected by any selector and is managed by storageGroups. Therefore, this EVS disk will be attached to the node as a raw disk and will not be initialized.

After the node is created, log in to the node and check whether a 100 GiB disk has been attached but not initialized.

```

[root@test-83790 ~]# lsblk -n
sda 8:0 0 50G 0 disk
└─sda1 8:1 0 50G 0 part /
sdb 8:16 0 100G 0 disk
├─vgpaas-dockersys 253:0 0 18G 0 lvm /var/lib/docker
├─vgpaas-thinpool_tmeta 253:1 0 3G 0 lvm
├─vgpaas-thinpool 253:3 0 67G 0 lvm
├─docker-253:0-786433-7cb37dc21202bfe2fc78dd1d33b70571e7e1982e56a4118f6facdc630cbc8b38 253:5 0 10G 0 dm /var/lib/docker/devicemapper/
├─docker-253:0-786433-e17cd8670b9f423eaff34b92bd82a2e620118227c26da2e41eda7894361c9942 253:6 0 10G 0 dm /var/lib/docker/devicemapper/
├─docker-253:0-786433-0dedb47e75eed3f635ce2d47c584587ae622c70dcb0eafeade9e14693a3146a0 253:7 0 10G 0 dm /var/lib/docker/devicemapper/
├─docker-253:0-786433-93ed7e6e14313d13ecfa1152937b153fe599c48cfdaf9ecd43c1c36cae89a38a 253:8 0 10G 0 dm /var/lib/docker/devicemapper/
├─docker-253:0-786433-e7066e08bf5c6249850a09e080cf43d9a7be499eae33aa8feb06c027d26fa1e9 253:9 0 10G 0 dm /var/lib/docker/devicemapper/
├─docker-253:0-786433-5ecc4420da9a58fb66108db599a8267af3e8856da86b9c3d7fb82099a8781ae8 253:10 0 10G 0 dm /var/lib/docker/devicemapper/
├─vgpaas-thinpool_tdata 253:2 0 67G 0 lvm
├─vgpaas-thinpool 253:3 0 67G 0 lvm
├─docker-253:0-786433-7cb37dc21202bfe2fc78dd1d33b70571e7e1982e56a4118f6facdc630cbc8b38 253:15 0 10G 0 dm /var/lib/docker/devicemapper/
├─docker-253:0-786433-e17cd8670b9f423eaff34b92bd82a2e620118227c26da2e41eda7894361c9942 253:16 0 10G 0 dm /var/lib/docker/devicemapper/
├─docker-253:0-786433-0dedb47e75eed3f635ce2d47c584587ae622c70dcb0eafeade9e14693a3146a0 253:17 0 10G 0 dm /var/lib/docker/devicemapper/
├─docker-253:0-786433-93ed7e6e14313d13ecfa1152937b153fe599c48cfdaf9ecd43c1c36cae89a38a 253:18 0 10G 0 dm /var/lib/docker/devicemapper/
├─docker-253:0-786433-e7066e08bf5c6249850a09e080cf43d9a7be499eae33aa8feb06c027d26fa1e9 253:19 0 10G 0 dm /var/lib/docker/devicemapper/
├─docker-253:0-786433-5ecc4420da9a58fb66108db599a8267af3e8856da86b9c3d7fb82099a8781ae8 253:10 0 10G 0 dm /var/lib/docker/devicemapper/
├─vgpaas-kubernetes 253:4 0 10G 0 lvm /mnt/paas/kubernetes/kubelet
sdc 8:32 0 100G 0 disk

```

The following is an API example:

```

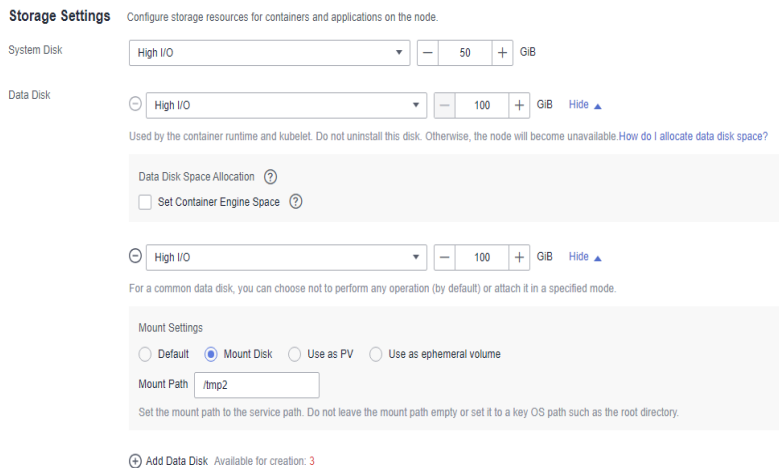
{
 "kind": "Node",
 "apiVersion": "v3",
 "metadata": {
 "name": "test-83790"
 },
 "spec": {
 "flavor": "c3.large.2",
 "az": "ap-southeast-1a",
 "os": "EulerOS 2.9",
 "dataVolumes": [
 {
 "size": 100,
 "volumetype": "SAS"
 },
 {
 "size": 100,
 "volumetype": "SAS"
 }
],
 "billingMode": 0,
 "extendParam": {
 "maxPods": 110
 },
 "nodeNicSpec": {
 "primaryNic": {

```

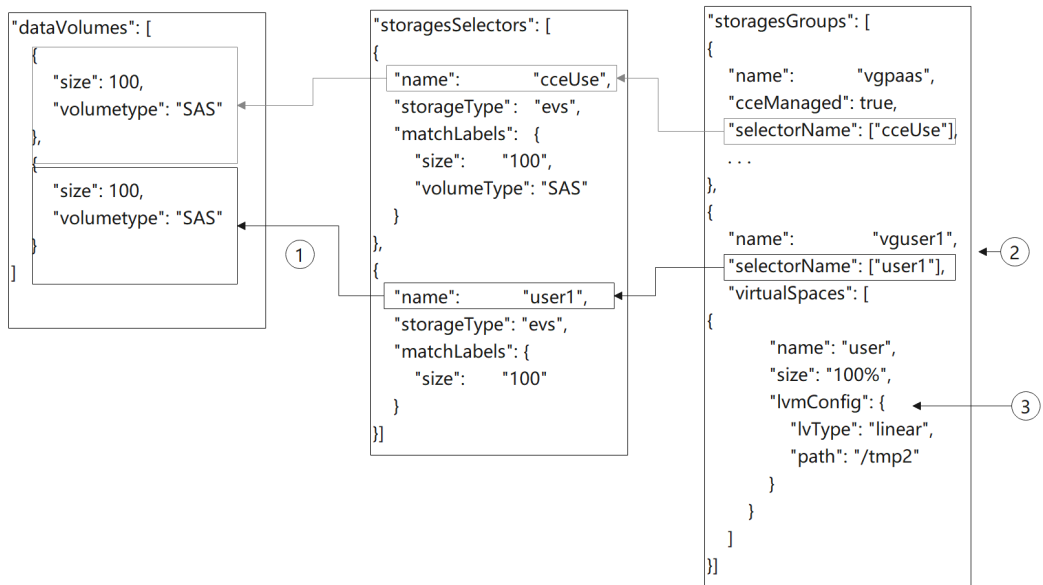
```
 "subnetId": "ca964acf-8468-4735-8229-97940ef6c881"
 }
},
"rootVolume": {
 "size": 50,
 "volumetype": "SAS"
},
"runtime": {
 "name": "docker"
},
"login": {
 "userPassword": {
 "username": "root",
 "password": "*****"
 }
},
"storage": {
 "storageSelectors": [
 {
 "name": "cceUse",
 "storageType": "evs",
 "matchLabels": {
 "size": "100",
 "volumeType": "SAS",
 "count": "1"
 }
 }
],
 "storageGroups": [
 {
 "name": "vgpaas",
 "selectorNames": [
 "cceUse"
],
 "cceManaged": true,
 "virtualSpaces": [
 {
 "name": "runtime",
 "size": "90%"
 },
 {
 "name": "kubernetes",
 "size": "10%"
 }
]
 }
]
},
"count": 1
}
```

## Attaching a Disk to a Specified Path

During node creation on the CCE console, click **Add Data Disk**. Then, click **Expand** next to the newly added data disk, select **Mount Disk**, and set the mount path. In this case, CCE initializes and attaches the disk by default.



The following figure shows the API calling logic.



1. The **user1** selector selects a 100 GiB data disk.
2. Create a VG named **vguser1** using LVM.
3. Strip all the space of **vguser1** into an LV named **user** and format the disk in ext4 format. Finally, attach the disk to the **/tmp2** directory.

After the node is created, log in to the node and check whether a 100 GiB disk has been attached and managed by LVM.



```

 "count": "1"
 }
 },
 {
 "name": "user1",
 "storageType": "evs",
 "matchLabels": {
 "size": "100",
 "volumeType": "SAS",
 "count": "1"
 }
 }
],
 "storageGroups": [
 {
 "name": "vgpaas",
 "selectorNames": [
 "cceUse"
],
 "cceManaged": true,
 "virtualSpaces": [
 {
 "name": "runtime",
 "size": "80%"
 },
 {
 "name": "kubernetes",
 "size": "20%"
 }
]
 }
],
 {
 "name": "vguser1",
 "selectorNames": [
 "user1"
],
 "virtualSpaces": [
 {
 "name": "user",
 "size": "100%",
 "lvmConfig": {
 "lvType": "linear",
 "path": "/tmp2"
 }
 }
]
 }
]
},
"count": 1
}
}

```

## Creating Striped LVs to Improve Disk Performance

Currently, the striped LV function is supported only by calling an API. The following is an example:



1. **storageSelectors** matches all EVS disks in **dataVolumes** because **matchLabels** is not contained in **storageSelectors**.
2. Create a VG named **vgpaas** using LVM.
3. Strip 90% of the **vgpaas** space into runtime LVs.
4. Strip 10% of the **vgpaas** space into Kubernetes LVs.

**NOTE**

- Two or more data disks are required for striping.
- When creating a striped LV, ensure that the types and sizes of the PVs added to the VG are the same. Otherwise, the creation will fail.
- When creating a striped LV, use the striping configuration for both the runtime LV and Kubernetes LV. Otherwise, the creation will fail.

Log in to the node and run the following command to view the striping result:

```
[root@test-83773 ~]# lvdisplay -m | grep -C 10 striped
LV Size 36.00 GiB
Current LE 9216
Segments 1
Allocation inherit
Read ahead sectors auto
- currently set to 512
Block device 253:0

--- Segments ---
Logical extents 0 to 9215:
 Type striped
 Stripes 2
 Stripe size 64.00 KiB
 Stripe 0:
 Physical volume /dev/sdb
 Physical extents 0 to 4607
 Stripe 1:
 Physical volume /dev/sdc
 Physical extents 0 to 4607

--
LV Size 20.00 GiB
Current LE 5120
Segments 1
Allocation inherit
Read ahead sectors auto
- currently set to 8192
Block device 253:4

--- Segments ---
Logical extents 0 to 5119:
 Type striped
 Stripes 2
 Stripe size 64.00 KiB
 Stripe 0:
 Physical volume /dev/sdb
```

The following is an API example:

```
{
 "kind": "Node",
 "apiVersion": "v3",
 "metadata": {
 "name": "test-83773"
 },
 "spec": {
 "flavor": "c3.large.2",
 "az": "ap-southeast-1a",
 "os": "EulerOS 2.9",
 "dataVolumes": [
 {
 "size": 100,
 "volumetype": "SAS"
 }
]
 }
}
```

```

 {
 "size": 100,
 "volumetype": "SAS"
 }
],
 "billingMode": 0,
 "extendParam": {
 "maxPods": 110
 },
 "nodeNicSpec": {
 "primaryNic": {
 "subnetId": "ca964acf-8468-4735-8229-97940ef6c881"
 }
 },
 "rootVolume": {
 "size": 50,
 "volumetype": "SAS"
 },
 "runtime": {
 "name": "docker"
 },
 "login": {
 "userPassword": {
 "username": "root",
 "password": "*****"
 }
 },
 "storage": {
 "storageSelectors": [
 {
 "name": "cceUse",
 "storageType": "evs"
 }
],
 "storageGroups": [
 {
 "name": "vgpaas",
 "selectorNames": [
 "cceUse"
],
 "cceManaged": true,
 "virtualSpaces": [
 {
 "name": "runtime",
 "size": "90%",
 "runtimeConfig": {
 "lvType": "striped"
 }
 },
 {
 "name": "kubernetes",
 "size": "10%",
 "lvmConfig": {
 "lvType": "striped"
 }
 }
]
 }
]
 },
 "count": 1
}

```