

Cloud Container Engine Autopilot

API Reference

Issue 01
Date 2025-01-03



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2025. 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.

Huawei Cloud Computing Technologies Co., Ltd.

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

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

Contents

| | |
|---|-----------|
| 1 Before You Start | 1 |
| 2 API Overview | 5 |
| 3 Calling APIs | 18 |
| 3.1 Making an API Request | 18 |
| 3.2 Authentication | 22 |
| 3.3 Response | 24 |
| 4 APIs | 26 |
| 4.1 Autopilot Cluster Management | 26 |
| 4.1.1 Creating a Cluster | 26 |
| 4.1.2 Obtaining a Cluster | 60 |
| 4.1.3 Obtaining Clusters in a Project | 78 |
| 4.1.4 Updating a Cluster | 97 |
| 4.1.5 Deleting a Cluster | 118 |
| 4.1.6 Obtaining a Cluster Certificate | 138 |
| 4.1.7 Obtaining Job Information | 147 |
| 4.1.8 Binding/Unbinding Public API Server Address | 154 |
| 4.1.9 Obtaining Cluster Access Address | 162 |
| 4.2 Add-on Management for Autopilot Clusters | 169 |
| 4.2.1 Installing an Add-on Instance | 169 |
| 4.2.2 Listing Add-on Templates | 187 |
| 4.2.3 Updating an Add-on Instance | 201 |
| 4.2.4 Rolling Back an Add-on Instance | 220 |
| 4.2.5 Deleting an Add-on Instance | 234 |
| 4.2.6 Obtaining an Add-on Instance | 238 |
| 4.2.7 Listing Add-on Instances | 250 |
| 4.3 Autopilot Cluster Upgrade | 262 |
| 4.3.1 Upgrading a Cluster | 262 |
| 4.3.2 Obtaining Cluster Upgrade Task Details | 275 |
| 4.3.3 Retrying a Cluster Upgrade Task | 281 |
| 4.3.4 Obtaining a List of Cluster Upgrade Task Details | 285 |
| 4.3.5 Performing a Pre-upgrade Check for a Cluster | 291 |
| 4.3.6 Obtaining Details About a Pre-upgrade Check Task of a Cluster | 304 |

| | |
|--|------------|
| 4.3.7 Obtaining a List of Pre-upgrade Check Tasks of a Cluster..... | 315 |
| 4.3.8 Performing a Post-upgrade Check for a Cluster..... | 326 |
| 4.3.9 Backing Up a Cluster..... | 333 |
| 4.3.10 Obtaining a List of Cluster Backup Task Details..... | 337 |
| 4.3.11 Obtaining the Cluster Upgrade Information..... | 343 |
| 4.3.12 Obtaining a Cluster Upgrade Path..... | 349 |
| 4.3.13 Obtaining the Configuration of Cluster Upgrade Feature Gates..... | 355 |
| 4.3.14 Enabling the Cluster Upgrade Process Booting Task..... | 360 |
| 4.3.15 Obtaining a List of Upgrade Workflows..... | 371 |
| 4.3.16 Obtaining Details About a Specified Cluster Upgrade Booting Task..... | 379 |
| 4.3.17 Updating the Status of a Specified Cluster Upgrade Booting Task..... | 388 |
| 4.4 Quota Management for Autopilot Clusters..... | 398 |
| 4.4.1 Obtaining CCE Resource Quotas..... | 398 |
| 4.5 Tag Management for Autopilot Clusters..... | 403 |
| 4.5.1 Adding Resource Tags to a Cluster in Batches..... | 404 |
| 4.5.2 Deleting Resource Tags from a Cluster in Batches..... | 411 |
| 4.6 Chart Management for Autopilot Clusters..... | 416 |
| 4.6.1 Uploading a Chart..... | 416 |
| 4.6.2 Obtaining a Chart List..... | 422 |
| 4.6.3 Obtaining a Release List..... | 427 |
| 4.6.4 Creating a Release..... | 433 |
| 4.6.5 Updating a Chart..... | 441 |
| 4.6.6 Deleting a Chart..... | 446 |
| 4.6.7 Updating a Release..... | 450 |
| 4.6.8 Obtaining a Chart..... | 458 |
| 4.6.9 Deleting a Release..... | 463 |
| 4.6.10 Obtaining a Release..... | 468 |
| 4.6.11 Downloading a Chart..... | 474 |
| 4.6.12 Obtaining Chart Values..... | 478 |
| 4.6.13 Obtaining Historical Records of a Release..... | 483 |
| 4.6.14 Obtaining the Quota of a User Chart..... | 489 |
| 5 Kubernetes APIs..... | 495 |
| 6 Permissions and Supported Actions..... | 500 |
| 7 Appendix..... | 507 |
| 7.1 Status Code..... | 507 |
| 7.2 Error Codes..... | 511 |
| 7.3 Obtaining a Project ID..... | 516 |
| 7.4 Obtaining an Account ID..... | 518 |
| 7.5 How to Obtain Parameters in the API URI..... | 518 |
| 7.6 Creating a VPC and Subnet..... | 520 |
| 7.7 Creating a Key Pair..... | 521 |

1 Before You Start

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.

If you plan to access CCE resources through APIs, 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 APIs: These APIs that can be accessed through the API gateway allow you to manage cloud service infrastructures (for example, creating a cluster). 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 cluster) 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 depending on Kubernetes-native APIs. Therefore, you are advised to use this type of APIs in unimportant scenarios, for example, short-term test clusters.

API Calling

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

Endpoints

An endpoint is the **request address** for calling an API. Endpoints vary depending on services and regions. For the endpoints of all services, see [Regions and Endpoints](#).

You need to select an endpoint based on your service requirements.

- The URL format for cluster and quota management is **https://Endpoint/uri**. In the URL, *uri* indicates the resource path, which is the path for API access.
- 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, which 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 . |

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.

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**

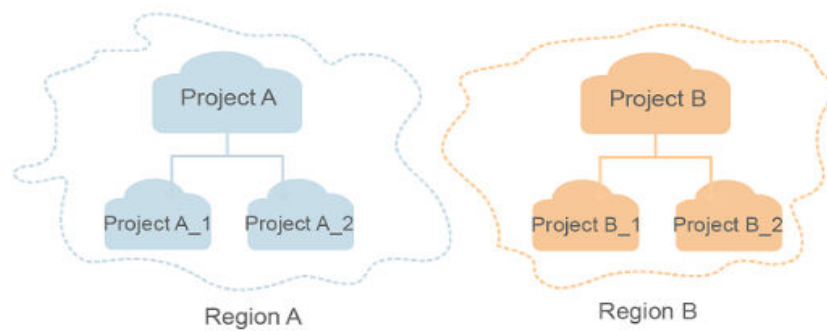
Regions are divided based on geographical location and network latency. Public services, such as Elastic Cloud Server (ECS), Elastic Volume Service (EVS), Object Storage Service (OBS), Virtual Private Cloud (VPC), Elastic IP (EIP), and Image Management Service (IMS), are shared within the same region. 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 grant 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 as proprietary APIs and Kubernetes-native APIs. Using these APIs, you can use all functions provided by CCE, including creating clusters, 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 management | Manage clusters, including creating and deleting clusters. You can use APIs in this category to create clusters and obtain information about created clusters. |
| | Add-on management | Manage add-ons, including querying add-on templates, and creating, updating, deleting, and obtaining add-on instances. |
| | Cluster upgrade | Upgrade clusters, including performing a pre-upgrade check and backing up cluster data. |
| | Quota management | Obtain quotas of CCE resources. |
| | Tag management | Manage tags, including adding and deleting resource tags of a cluster. |
| | Chart management | Manage charts, including creating, updating, obtaining, and deleting charts and chart releases. |
| Kubernetes-native APIs | None | For details about how to call Kubernetes-native APIs, see Kubernetes APIs . |

Kubernetes APIs

| API | Function | URI |
|-----------------|--|---|
| Node | Obtaining 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} |
| 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 |
| | Obtaining a specified pod | GET /api/v1/namespaces/{namespace}/pods/{name} |

| API | Function | URI |
|-------------|---|--|
| | 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} |
| Deploy ment | Creating a Deployment | POST /apis/apps/v1/namespaces/{namespace}/deployments |
| | Rolling back a Deployment | PATCH /apis/apps/v1/namespaces/{namespace}/deployments/{name} |
| | Deleting a Deployment | DELETE /apis/apps/v1/namespaces/{namespace}/deployments/{name} |
| | Deleting all Deployments | DELETE /apis/apps/v1/namespaces/{namespace}/deployments |
| | Obtaining a specified Deployment | GET /apis/apps/v1/namespaces/{namespace}/deployments/{name} |
| | Obtaining the status of a specified Deployment | GET /apis/apps/v1/namespaces/{namespace}/deployments/{name}/status |
| | Obtaining 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 |

| API | Function | URI |
|--------------|--|---|
| | Updating the scaling operation of a specified Deployment | PATCH /apis/apps/v1/namespaces/{namespace}/deployments/{name}/scale |
| Stateful Set | 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 |
| | Obtaining a specified StatefulSet | GET /apis/apps/v1/namespaces/{namespace}/statefulsets/{name} |
| | Obtaining 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} |
| | 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 |
| Job | Creating a job | POST /apis/batch/v1/namespaces/{namespace}/jobs |
| | Deleting a job | DELETE /apis/batch/v1/namespaces/{namespace}/jobs/{name} |
| | Deleting all jobs | DELETE /apis/batch/v1/namespaces/{namespace}/jobs |
| | Obtaining a specified job | GET /apis/batch/v1/namespaces/{namespace}/jobs/{name} |
| | Obtaining 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 |

| API | Function | URI |
|-----------------------|--|---|
| | 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/v1/namespaces/{namespace}/cronjobs |
| | Deleting a cron job | DELETE /apis/batch/v1/namespaces/{namespace}/cronjobs/{name} |
| | Deleting all cron jobs | DELETE /apis/batch/v1/namespaces/{namespace}/cronjobs |
| | Obtaining a specified cron job | GET /apis/batch/v1/namespaces/{namespace}/cronjobs/{name} |
| | Obtaining the status of a specified cron job | GET /apis/batch/v1/namespaces/{namespace}/cronjobs/{name}/status |
| | Replacing a specified cron job | PUT /apis/batch/v1/namespaces/{namespace}/cronjobs/{name} |
| | Replacing the status of a specified cron job | PUT /apis/batch/v1/namespaces/{namespace}/cronjobs/{name}/status |
| | Listing cron jobs in a specified namespace | GET /apis/batch/v1/namespaces/{namespace}/cronjobs |
| | Listing all cron jobs | GET /apis/batch/v1/cronjobs |
| | Updating the status of a specified cron job | PATCH /apis/batch/v1/namespaces/{namespace}/cronjobs/{name}/status |
| | Updating a specified cron job | PATCH /apis/batch/v1/namespaces/{namespace}/cronjobs/{name} |
| ReplicaSet | Listing ReplicaSets | GET /apis/apps/v1/namespaces/{namespace}/replicasets |
| | Obtaining 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} |

| API | Function | URI |
|-----------|---|---|
| | Deleting all ReplicationControllers | DELETE /api/v1/namespaces/{namespace}/replicationcontrollers |
| | Obtaining 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} |
| | 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 |

| API | Function | URI |
|---------|---|---|
| | 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 |
| | Updating a specified ingress | PATCH /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name} |
| | Replacing a specified ingress | PUT /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name} |
| | Deleting an ingress | DELETE /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name} |
| | Deleting all ingresses | DELETE /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses |
| | Obtaining a specified ingress | GET /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name} |
| | Listing ingresses in a specified namespace | GET /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses |
| | Listing ingresses | GET /apis/networking.k8s.io/v1/ingresses |
| | Obtaining the status of an ingress in a specified namespace | GET /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name}/status |
| | Replacing the status of an ingress in a specified namespace | PUT /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name}/status |
| | Updating the status of an ingress in a specified namespace | PATCH /apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses/{name}/status |

| API | Function | URI |
|------------------|--|---|
| 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 |
| | Obtaining 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 |
| | Obtaining a specified PersistentVolume | GET /api/v1/persistentvolumes/{name} |
| | 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} |

| API | Function | URI |
|-----------------------|---|---|
| PersistentVolumeClaim | Creating a PersistentVolumeClaim | POST /api/v1/namespaces/{namespace}/persistentvolumeclaims |
| | Deleting a specified PersistentVolumeClaim | DELETE /api/v1/namespaces/{namespace}/persistentvolumeclaims/{name} |
| | Deleting all PersistentVolumeClaims | DELETE /api/v1/namespaces/{namespace}/persistentvolumeclaims |
| | Obtaining 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 PersistentVolumeClaims in a specified namespace | GET /api/v1/namespaces/{namespace}/persistentvolumeclaims |
| | Listing all PersistentVolumeClaims | 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 |
| | Obtaining 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} |

| API | Function | URI |
|-----------------------------|---|--|
| 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 |
| | Obtaining 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 |
| | Obtaining a specified ClusterRole | GET /apis/rbac.authorization.k8s.io/v1/clusterroles/{name} |
| | Listing ClusterRoles | GET /apis/rbac.authorization.k8s.io/v1/clusterroles |
| 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} |

| API | Function | URI |
|--------------------------|--|--|
| | Deleting ClusterRoleBindings in batches | DELETE /apis/rbac.authorization.k8s.io/v1/clusterrolebindings |
| | Obtaining 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 |
| | Obtaining 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/ RoleBin ding | 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} |
| | 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 |
| | Obtaining a specified RoleBinding | GET /apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings/{name} |

| API | Function | URI |
|------------|---|--|
| | 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 |
| | 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 |
| | 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 |

| API | Function | URI |
|-------|--|---|
| | 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 | Obtaining 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 how a REST API is formed and uses the IAM [API for obtaining a user token](#) as an example to describe how to call an API. The obtained token can be used to authenticate the calls 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.

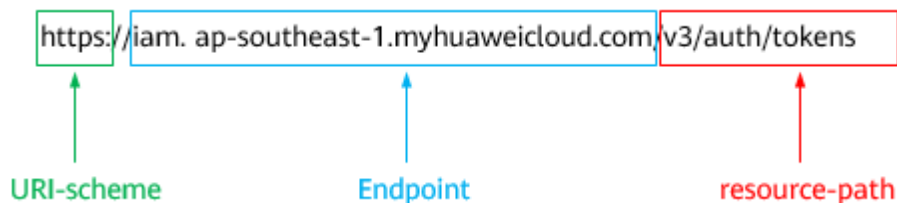
Table 3-1 URI parameters

| 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 URI of the API for **obtaining a user token**, the request method is **POST**, and 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. The user token is a response to the API used to obtain a user token . This API is the only one 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 request headers 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 for [obtaining a user token](#), the request parameters and parameter description can be obtained from the API request. The following provides an example request with the body included. Replace *username*, *domainname*, ******* (login password), and *xxxxxxxxxxxxxxxxxxxx* (project name) with the actual values. To learn how to obtain a project ID, see [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.

When calling the API to **obtain a user token**, you must 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 the calling of the API for [obtaining 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 of the API for [obtaining 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 of the API for [obtaining 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 Autopilot Cluster Management

4.1.1 Creating a Cluster

Function

This API is used to create an empty cluster, which has only master nodes but no worker nodes.

NOTE

- The URL for cluster management is in the format of **https://Endpoint/uri**, where **uri** specifies the resource path for API access.

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, manageable virtual network environment for CCE clusters. If no VPC is available, you need to create one first. For details, see [Creating a VPC](#).
- Before creating a cluster, you should properly plan the Service CIDR block. After a cluster using a container tunnel network is created, you cannot modify the CIDR blocks. After a cluster using a VPC network or Cloud Native 2.0 network is created, you can only add CIDR blocks or subnet CIDR blocks to the cluster but cannot modify the existing CIDR blocks or subnet CIDR blocks. To modify the existing CIDR blocks, you need to create a new cluster. Exercise caution when performing this operation.
- 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 to see if there is any agency. 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 /autopilot/v3/projects/{project_id}/clusters

Table 4-1 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| project_id | Yes | String | Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI . Constraints: None Options: Project IDs of the account Default value: N/A |

Request Parameters

Table 4-2 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Table 4-3 Request body parameters

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|---|
| kind | Yes | String | API type. The value is fixed at Cluster or cluster and cannot be changed. |

| Parameter | Mandatory | Type | Description |
|------------|-----------|---|--|
| apiVersion | Yes | String | API version. The value is fixed at v3 and cannot be changed. |
| metadata | Yes | AutopilotClusterMetadata object | Basic information about a cluster. Metadata is a collection of attributes. |
| spec | Yes | AutopilotClusterSpec object | Element type of the collection class. The main part of a cluster object to be managed is in spec . CCE creates or updates objects by defining or updating spec . |
| status | No | AutopilotClusterStatus object | Element type of the collection class, which is used to record the statuses of an object in the system, including the cluster status and the ID of the cluster creation job |

Table 4-4 AutopilotClusterMetadata

| 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 resource. The value is automatically generated after the object is created. A custom value will not take effect. When you create a yearly/monthly cluster, no cluster ID will be returned in the response body. |

| Parameter | Mandatory | Type | Description |
|-------------|-----------|--------------------|---|
| alias | No | 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 is left blank, the alias of the cluster is the same as the cluster name. In the response body for creating a cluster, if the cluster alias is not configured, the cluster alias will not be returned.</p> |
| 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 are not used to 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. |
| 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 a cluster during the upgrade. Customized values are invalid.</p> |

| Parameter | Mandatory | Type | Description |
|-----------------------|-----------|--------|---------------------------------|
| creationTimes tamp | No | String | Time when a cluster was created |
| updateTimest amp | No | String | Time when a cluster was updated |

Table 4-5 AutopilotClusterSpec

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|---|
| category | No | String | Cluster type. Only Turbo is supported. |
| type | No | String | Master node architecture. <ul style="list-style-type: none"> • VirtualMachine: The master node is an x86 server. |
| flavor | Yes | String | Cluster specifications: cce.autopilot.cluster |
| version | No | String | Version of a cluster, which mirrors the baseline version of the Kubernetes community. The latest version is recommended. You can create clusters of the latest three versions on the CCE console. To learn which cluster versions are available, log in to the CCE console, create a cluster, and check supported cluster versions. 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. NOTE <ul style="list-style-type: none"> • If not specified, a cluster of the latest version will be created. |

| Parameter | Mandatory | Type | Description |
|-----------------|-----------|--------|--|
| platformVersion | No | String | <p>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. This parameter cannot be customized, and when you create a cluster, the latest corresponding platform version is selected automatically.</p> <p>The format of platformVersion is cce.X.Y.</p> <ul style="list-style-type: none"> • X: internal feature version, which indicates 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 released. 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 blank. To modify cluster description after a cluster is created, call the API for updating a specified cluster 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>The custom SAN field in the API server certificate of a cluster. It complies with the format specifications defined by the SSL standard X509. It is not supported by Autopilot clusters.</p> <ol style="list-style-type: none"> 1. Duplicate names are not allowed. 2. The format 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> |
| enableSnat | No | Boolean | <p>Whether SNAT is configured for a cluster. This parameter is used and returned only by the Autopilot cluster creation API. After this function is enabled, the cluster can access the Internet through a NAT gateway. By default, the existing NAT gateway in the selected VPC is used. Otherwise, the system automatically creates a NAT gateway of the default specifications, binds an EIP to the NAT gateway, and configures SNAT rules.</p> |
| enableSWRImageAccess | No | Boolean | <p>Whether a cluster is interconnected with SWR. This parameter is used and returned only by the Autopilot cluster creation API. To ensure that your cluster nodes can pull images from SWR, the existing SWR and OBS endpoints in the selected VPC are used by default. If not, new SWR and OBS endpoints will be automatically created.</p> |
| enableAutopilot | No | Boolean | <p>Whether the cluster is an Autopilot cluster</p> |

| Parameter | Mandatory | Type | Description |
|----------------------|-----------|---|---|
| ipv6enable | No | Boolean | Whether a cluster uses the IPv6 mode. This parameter is not supported by Autopilot clusters. |
| hostNetwork | Yes | AutopilotHostNetwork object | Node networking parameters, including VPC and subnet ID. This field is mandatory because nodes in a cluster communicate with each other using a VPC. |
| containerNetwork | Yes | AutopilotContainerNetwork object | Container networking parameters, including the container network model and container CIDR block |
| eniNetwork | No | AutopilotEniNetwork object | Cloud Native 2.0 network settings |
| serviceNetwork | No | AutopilotServiceNetwork object | Service CIDR block, including IPv4 CIDR block |
| authentication | No | AutopilotAuthentication object | Configuration of the cluster authentication mode, which is not supported by Autopilot clusters |
| billingMode | No | Integer | Cluster billing mode <ul style="list-style-type: none"> • 0: pay-per-use This parameter defaults to pay-per-use . |
| kubernetesSvcIpRange | No | String | Service CIDR block or the IP address range which the kubernetes clusterIP must fall within. If this parameter is not specified during cluster creation, the default value 10.247.0.0/16 is used. This parameter is being deprecated, so you can use serviceNetwork instead. The new field contains the IPv4 CIDR blocks. |
| clusterTags | No | Array of AutopilotResourceTag objects | Cluster resource tags |

| Parameter | Mandatory | Type | Description |
|------------------------|-----------|--|--|
| kubeProxyMode | No | String | <p>Service forwarding mode.</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. <p>NOTE iptables is used by default.</p> |
| az | No | String | <p>AZ. This field is returned only for a query.</p> <p>For details about AZs supported by CCE, see Regions and Endpoints.</p> |
| extendParam | No | AutopilotClusterExtendParam object | <p>Extended field to decide whether a cluster will span across AZs or belong to a specified enterprise project, or whether a dedicated CCE cluster is to be created</p> |
| configurationsOverride | No | Array of AutopilotPackageConfiguration objects | <p>Overrides the default component configurations in a cluster. It is not supported in CCE Autopilot clusters.</p> |

Table 4-6 AutopilotHostNetwork

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|--|
| vpc | Yes | String | <p>ID of the VPC used to create a master node.</p> <p>Obtain the value using either of the following methods:</p> <ul style="list-style-type: none"> Method 1: Log in to the VPC console and view the VPC ID on the VPC details page. Method 2: Use the VPC API. For details, see Querying VPCs. |
| subnet | Yes | String | <p>Network ID of the subnet used to create a master node.</p> <p>Obtain the value using either of the following methods:</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. For details, see Querying Subnets. |

Table 4-7 AutopilotContainerNetwork

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|---|
| mode | Yes | String | <p>Container network type</p> <ul style="list-style-type: none"> eni*: specifies the Cloud Native 2.0 network model. This model has integrated cloud native elastic network interfaces (ENIs), uses VPC CIDR blocks to allocate container IP addresses, and supports ELB passthrough networking for high network performance. Use this model when creating a cluster. |

Table 4-8 AutopilotEniNetwork

| Parameter | Mandatory | Type | Description |
|-----------|-----------|---|---|
| subnets | Yes | Array of AutopilotNetworkSubnet objects | <p>IPv4 subnet ID list of the subnet where an ENI resides. Obtain the value using either of the following methods:</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 4-9 AutopilotNetworkSubnet

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|---|
| subnetID | Yes | String | <p>Specifies the IPv4 subnet ID of the subnet used to create control nodes and containers. (IPv6 is not supported.) To obtain the subnet ID, use either of the following methods:</p> <ul style="list-style-type: none"> Method 1: Log in to the VPC console and click a subnet in the VPC. On the subnet details page, search for the IPv4 subnet ID. Method 2: Use the VPC API for obtaining subnets. <p>For details, see Querying Subnets.</p> |

Table 4-10 AutopilotServiceNetwork

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

Table 4-11 AutopilotAuthentication

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|---|
| mode | No | String | Cluster authentication mode. The default value is rbac . |

Table 4-12 AutopilotResourceTag

| 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>. |

Table 4-13 AutopilotClusterExtendParam

| Parameter | Mandatory | Type | Description |
|---------------------|-----------|--------|--|
| enterpriseProjectId | No | String | ID of the enterprise project that a cluster belongs to NOTE <ul style="list-style-type: none"> An enterprise project can be configured only after the enterprise project function is enabled. |
| upgradeFrom | No | String | Records of how a cluster is upgraded to its current version |

Table 4-14 AutopilotPackageConfiguration

| Parameter | Mandatory | Type | Description |
|----------------|-----------|---|-------------------------------|
| name | No | String | Component name |
| configurations | No | Array of AutopilotConfigurationItem objects | Component configuration items |

Table 4-15 AutopilotConfigurationItem

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

Table 4-16 AutopilotClusterStatus

| 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 and needs to be manually deleted. ● 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. ● ResizeFailed: Cluster specification changing is abnormal. ● 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. ● Freezing: The cluster is being frozen. ● Frozen: The cluster has been frozen. ● UnFreezing: The cluster is being unfrozen. ● Awaking: The cluster is being woken up from hibernation. ● Empty: The cluster does not have any resources. This field is discarded. |

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--|---|
| | | | <ul style="list-style-type: none"> Error: Resources in the cluster are abnormal. You can manually delete the cluster. |
| jobID | No | String | <p>ID of the task associated with the current cluster status.</p> <p>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 being 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 | Why a cluster changes to the current state. This parameter is returned if the cluster is not in the Available state. |
| message | No | String | Detailed information about why a cluster changes to the current state. This parameter is returned if the cluster is not in the Available state. |
| endpoints | No | Array of AutopilotClusterEndpoint s objects | Access address of kube-apiserver in a cluster |
| isLocked | No | Boolean | CBC resource locked |
| lockScene | No | String | Scenario where the CBC resource is locked |
| lockSource | No | String | Resource locked |
| lockSourceId | No | String | ID of a locked resource |

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| deleteOption | No | Object | Status of the configuration to be deleted. This parameter is contained only in the response to the deletion request. |
| deleteStatus | No | Object | Information of the status to be deleted. This parameter is contained only in the response to the deletion request. |

Table 4-17 AutopilotClusterEndpoints

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|---|
| url | No | String | Access address of kube-apiserver in a 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 4-18 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 | AutopilotCluster-Metadata object | Basic information about a cluster. Metadata is a collection of attributes. |
| spec | AutopilotCluster-Spec object | Element type of the collection class. The main part of a cluster object to be managed is in spec . CCE creates or updates objects by defining or updating spec . |

| Parameter | Type | Description |
|-----------|--------------------------------------|--|
| status | AutopilotClusterStatus object | Element type of the collection class, which is used to record the statuses of an object in the system, including the cluster status and the ID of the cluster creation job |

Table 4-19 AutopilotClusterMetadata

| 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 resource. The value is automatically generated after the object is created. A custom value will not 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 is left blank, the alias of the cluster is the same as the cluster name. In the response body for creating a cluster, if the cluster alias is not configured, the cluster alias will not be returned. |

| Parameter | Type | Description |
|-------------------|--------------------|---|
| 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 are not used to 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. |
| labels | Map<String,String> | Cluster labels in the format of key-value pairs. <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 a cluster during the upgrade. Customized values are invalid.</p> |
| creationTimestamp | String | Time when a cluster was created |
| updateTimestamp | String | Time when a cluster was updated |

Table 4-20 AutopilotClusterSpec

| Parameter | Type | Description |
|-----------|--------|--|
| category | String | Cluster type. Only Turbo is supported. |
| type | String | Master node architecture. <ul style="list-style-type: none"> VirtualMachine: The master node is an x86 server. |
| flavor | String | Cluster specifications: cce.autopilot.cluster |

| Parameter | Type | Description |
|-----------------|--------|--|
| version | String | <p>Version of a cluster, which mirrors the baseline version of the Kubernetes community. The latest version is recommended.</p> <p>You can create clusters of the latest three versions on the CCE console. To learn which cluster versions are available, log in to the CCE console, create a cluster, and check supported cluster versions.</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. |
| platformVersion | String | <p>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. This parameter cannot be customized, and when you create a cluster, the latest corresponding platform version is selected automatically.</p> <p>The format of platformVersion is cce.X.Y.</p> <ul style="list-style-type: none"> X: internal feature version, which indicates 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 released. 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 blank. To modify cluster description after a cluster is created, call the API for updating a specified cluster or go to the cluster details page on the CCE console. Only UTF-8 encoding is supported. |
| customSan | Array of strings | The custom SAN field in the API server certificate of a cluster. It complies with the format specifications defined by the SSL standard X509. It is not supported by Autopilot clusters. <ol style="list-style-type: none"> 1. Duplicate names are not allowed. 2. The format 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 |
| enableSnat | Boolean | Whether SNAT is configured for a cluster. This parameter is used and returned only by the Autopilot cluster creation API. After this function is enabled, the cluster can access the Internet through a NAT gateway. By default, the existing NAT gateway in the selected VPC is used. Otherwise, the system automatically creates a NAT gateway of the default specifications, binds an EIP to the NAT gateway, and configures SNAT rules. |
| enableSWRImage Access | Boolean | Whether a cluster is interconnected with SWR. This parameter is used and returned only by the Autopilot cluster creation API. To ensure that your cluster nodes can pull images from SWR, the existing SWR and OBS endpoints in the selected VPC are used by default. If not, new SWR and OBS endpoints will be automatically created. |
| enableAutopilot | Boolean | Whether the cluster is an Autopilot cluster |

| Parameter | Type | Description |
|----------------------|---|---|
| ipv6enable | Boolean | Whether a cluster uses the IPv6 mode. This parameter is not supported by Autopilot clusters. |
| hostNetwork | AutopilotHostNetwork object | Node networking parameters, including VPC and subnet ID. This field is mandatory because nodes in a cluster communicate with each other using a VPC. |
| containerNetwork | AutopilotContainerNetwork object | Container networking parameters, including the container network model and container CIDR block |
| eniNetwork | AutopilotEniNetwork object | Cloud Native 2.0 network settings |
| serviceNetwork | AutopilotServiceNetwork object | Service CIDR block, including IPv4 CIDR block |
| authentication | AutopilotAuthentication object | Configuration of the cluster authentication mode, which is not supported by Autopilot clusters |
| billingMode | Integer | Cluster billing mode <ul style="list-style-type: none"> • 0: pay-per-use This parameter defaults to pay-per-use . |
| kubernetesSvcIpRange | String | Service CIDR block or the IP address range which the kubernetes clusterIP must fall within. If this parameter is not specified during cluster creation, the default value 10.247.0.0/16 is used. This parameter is being deprecated, so you can use serviceNetwork instead. The new field contains the IPv4 CIDR blocks. |
| clusterTags | Array of AutopilotResourceTag objects | Cluster resource tags |

| Parameter | Type | Description |
|-----------------------------|--|---|
| kubeProxyMode | String | Service forwarding mode. <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. <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 | AutopilotCluster-ExtendParam object | Extended field to decide whether a cluster will span across AZs or belong to a specified enterprise project, or whether a dedicated CCE cluster is to be created |
| configurationsO- verride | Array of AutopilotPackageConfiguration objects | Overrides the default component configurations in a cluster. It is not supported in CCE Autopilot clusters. |

Table 4-21 AutopilotHostNetwork

| Parameter | Type | Description |
|-----------|--------|---|
| vpc | String | ID of the VPC used to create a master node. Obtain the value using either of the following methods: <ul style="list-style-type: none"> Method 1: Log in to the VPC console and view the VPC ID on the VPC details page. Method 2: Use the VPC API. For details, see Querying VPCs. |

| Parameter | Type | Description |
|-----------|--------|--|
| subnet | String | <p>Network ID of the subnet used to create a master node. Obtain the value using either of the following methods:</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> |

Table 4-22 AutopilotContainerNetwork

| Parameter | Type | Description |
|-----------|--------|---|
| mode | String | <p>Container network type</p> <ul style="list-style-type: none"> eni*: specifies the Cloud Native 2.0 network model. This model has integrated cloud native elastic network interfaces (ENIs), uses VPC CIDR blocks to allocate container IP addresses, and supports ELB passthrough networking for high network performance. Use this model when creating a cluster. |

Table 4-23 AutopilotEniNetwork

| Parameter | Type | Description |
|-----------|---|---|
| subnets | Array of AutopilotNetworkSubnet objects | <p>IPv4 subnet ID list of the subnet where an ENI resides. Obtain the value using either of the following methods:</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 4-24 AutopilotNetworkSubnet

| Parameter | Type | Description |
|-----------|--------|---|
| subnetID | String | <p>Specifies the IPv4 subnet ID of the subnet used to create control nodes and containers. (IPv6 is not supported.) To obtain the subnet ID, use either of the following methods:</p> <ul style="list-style-type: none"> Method 1: Log in to the VPC console and click a subnet in the VPC. On the subnet details page, search for the IPv4 subnet ID. Method 2: Use the VPC API for obtaining subnets. <p>For details, see Querying Subnets.</p> |

Table 4-25 AutopilotServiceNetwork

| Parameter | Type | Description |
|-----------|--------|---|
| IPv4CIDR | String | <p>Value range of the Kubernetes clusterIP IPv4 CIDR block. If this parameter is not specified during cluster creation, the default value 10.247.0.0/16 is used.</p> |

Table 4-26 AutopilotAuthentication

| Parameter | Type | Description |
|-----------|--------|---|
| mode | String | <p>Cluster authentication mode. The default value is rbac.</p> |

Table 4-27 AutopilotResourceTag

| Parameter | Type | Description |
|-----------|--------|---|
| key | String | <p>Key.</p> <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 _sys_. |

| Parameter | Type | Description |
|-----------|--------|--|
| 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 4-28 AutopilotClusterExtendParam

| Parameter | Type | Description |
|---------------------|--------|--|
| enterpriseProjectId | String | ID of the enterprise project that a cluster belongs to NOTE <ul style="list-style-type: none"> • An enterprise project can be configured only after the enterprise project function is enabled. |
| upgradeFrom | String | Records of how a cluster is upgraded to its current version |

Table 4-29 AutopilotPackageConfiguration

| Parameter | Type | Description |
|----------------|---|-------------------------------|
| name | String | Component name |
| configurations | Array of AutopilotConfigurationItem objects | Component configuration items |

Table 4-30 AutopilotConfigurationItem

| Parameter | Type | Description |
|-----------|--------|------------------------------------|
| name | String | Component configuration item name |
| value | Object | Component configuration item value |

Table 4-31 AutopilotClusterStatus

| 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 and needs to be manually deleted. ● 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. ● ResizeFailed: Cluster specification changing is abnormal. ● 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. ● Freezing: The cluster is being frozen. ● Frozen: The cluster has been frozen. ● UnFreezing: The cluster is being unfrozen. ● 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. You can manually delete the cluster. |

| Parameter | Type | Description |
|--------------|---|--|
| jobID | String | <p>ID of the task associated with the current cluster status. 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 being 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 | Why a cluster changes to the current state. This parameter is returned if the cluster is not in the Available state. |
| message | String | Detailed information about why a cluster changes to the current state. This parameter is returned if the cluster is not in the Available state. |
| endpoints | Array of AutopilotCluster-Endpoints objects | Access address of kube-apiserver in a cluster |
| isLocked | Boolean | CBC resource locked |
| lockScene | String | Scenario where the CBC resource is locked |
| lockSource | String | Resource locked |
| lockSourceId | String | ID of a locked resource |
| deleteOption | Object | Status of the configuration to be deleted. This parameter is contained only in the response to the deletion request. |
| deleteStatus | Object | Information of the status to be deleted. This parameter is contained only in the response to the deletion request. |

Table 4-32 AutopilotClusterEndpoints

| Parameter | Type | Description |
|-----------|--------|---|
| url | String | Access address of kube-apiserver in a 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 Autopilot cluster of v1.28 billed on a pay-per-use basis.

```

/autopilot/v3/projects/{project_id}/clusters

{
  "kind": "Cluster",
  "apiVersion": "v3",
  "metadata": {
    "name": "test-cluster-autopilot",
    "annotations": {
      "cluster.install.addons/install": "[{"addonTemplateName":"coredns","values":{"flavor":{"category":"Autopilot"},"is_default":true,"name":"autopilot-flavor1","replicas":2,"resources":{"limitsCpu":"1","limitsMem":"2Gi"},"name":"coredns","requestsCpu":"1","requestsMem":"2Gi"}}], [{"addonTemplateName":"metrics-server","values":{"flavor":{"category":"Autopilot"},"description":"custom resources in autopilot cluster","is_default":true,"name":"autopilot-flavor1","replicas":2,"resources":{"limitsCpu":"1","limitsMem":"2Gi"},"name":"metrics-server","requestsCpu":"1","requestsMem":"2Gi"}]"}]
    }
  },
  "spec": {
    "category": "Turbo",
    "flavor": "cce.autopilot.cluster",
    "type": "VirtualMachine",
    "version": "v1.28",
    "hostNetwork": {
      "vpc": "c6549063-d459-4ae1-9550-b5fec6741b0f",
      "subnet": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
    },
    "extendParam": {
      "enterpriseProjectId": "0"
    },
    "containerNetwork": {
      "mode": "eni"
    },
    "description": "",
    "billingMode": 0,
    "eniNetwork": {
      "subnets": [ {
        "subnetID": "186f9322-50c5-4e5a-91e3-47da86959afc"
      } ]
    },
    "enableSWRIImageAccess": true,
    "enableSnat": true,
    "serviceNetwork": {
      "IPv4CIDR": "10.247.0.0/16"
    }
  }
}

```

Example Responses

Status code: 201

The cluster creation job is delivered.

```
{
  "kind": "Cluster",
  "apiVersion": "v3",
  "metadata": {
    "name": "test-cluster-autopilot",
    "uid": "e18f8b25-2270-11ef-a160-0255ac100100",
    "creationTimestamp": "2024-06-04 12:49:28.773718231 +0000 UTC",
    "updateTimestamp": "2024-06-04 12:49:28.773718305 +0000 UTC",
    "annotations": {
      "jobid": "e1c49157-2270-11ef-a160-0255ac100100",
      "resourceJobId": "e18fa26f-2270-11ef-a160-0255ac100100"
    }
  },
  "spec": {
    "category": "Turbo",
    "type": "VirtualMachine",
    "flavor": "cce.autopilot.cluster",
    "version": "v1.28",
    "platformVersion": "cce.4.0",
    "hostNetwork": {
      "vpc": "c6549063-d459-4ae1-9550-b5fec6741b0f",
      "subnet": "3b18c2d5-b352-4f59-b421-c2d2d48a1333"
    },
    "containerNetwork": {
      "mode": "eni"
    },
    "eniNetwork": {
      "subnets": [ {
        "subnetID": "186f9322-50c5-4e5a-91e3-47da86959afc"
      } ]
    },
    "serviceNetwork": {
      "IPv4CIDR": "10.247.0.0/16"
    },
    "authentication": {
      "mode": "rbac"
    },
    "billingMode": 0,
    "kubernetesSvcIpRange": "10.247.0.0/16",
    "kubeProxyMode": "iptables",
    "extendParam": {
      "enterpriseProjectId": "0"
    },
    "enableSWRIImageAccess": true,
    "enableSnat": true
  },
  "status": {
    "phase": "Creating",
    "jobID": "e1c49157-2270-11ef-a160-0255ac100100"
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Create an Autopilot cluster of v1.28 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;
import java.util.Map;
import java.util.HashMap;

public class CreateAutopilotClusterSolution {

    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();
        CreateAutopilotClusterRequest request = new CreateAutopilotClusterRequest();
        AutopilotCluster body = new AutopilotCluster();
        AutopilotClusterExtendParam extendParamSpec = new AutopilotClusterExtendParam();
        extendParamSpec.withEnterpriseProjectId("0");
        AutopilotServiceNetwork serviceNetworkSpec = new AutopilotServiceNetwork();
        serviceNetworkSpec.withIpv4CIDR("10.247.0.0/16");
        List<AutopilotNetworkSubnet> listEniNetworkSubnets = new ArrayList<>();
        listEniNetworkSubnets.add(
            new AutopilotNetworkSubnet()
                .withSubnetID("186f9322-50c5-4e5a-91e3-47da86959afc")
        );
        AutopilotEniNetwork eniNetworkSpec = new AutopilotEniNetwork();
        eniNetworkSpec.withSubnets(listEniNetworkSubnets);
        AutopilotContainerNetwork containerNetworkSpec = new AutopilotContainerNetwork();
        containerNetworkSpec.withMode(AutopilotContainerNetwork.ModeEnum.fromValue("eni"));
        AutopilotHostNetwork hostNetworkSpec = new AutopilotHostNetwork();
        hostNetworkSpec.withVpc("c6549063-d459-4ae1-9550-b5fec6741b0f")
            .withSubnet("xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx");
        AutopilotClusterSpec specbody = new AutopilotClusterSpec();
        specbody.withCategory(AutopilotClusterSpec.CategoryEnum.fromValue("Turbo"))
            .withType(AutopilotClusterSpec.TypeEnum.fromValue("VirtualMachine"))
            .withFlavor("cce.autopilot.cluster")
            .withVersion("v1.28")
            .withDescription("")
            .withEnableSnat(true)
            .withEnableSWRImageAccess(true)
            .withHostNetwork(hostNetworkSpec)
            .withContainerNetwork(containerNetworkSpec)
            .withEniNetwork(eniNetworkSpec)
            .withServiceNetwork(serviceNetworkSpec)
            .withBillingMode(0)
            .withExtendParam(extendParamSpec);
        Map<String, String> listMetadataAnnotations = new HashMap<>();
        listMetadataAnnotations.put("cluster.install.addons/install",
```

```
    [{"addonTemplateName":"coredns","values":{"flavor":{"category":
["Autopilot"],"is_default":true,"name":"autopilot-flavor1","replicas":2,"resources":
[{"limitsCpu":1,"limitsMem":"2Gi","name":"coredns","requestsCpu":1,"requestsMem":"2Gi"}]}},
{"addonTemplateName":"metrics-server","values":{"flavor":{"category":["Autopilot"],"description":"custom
resources in autopilot cluster","is_default":true,"name":"autopilot-flavor1","replicas":2,"resources":
[{"limitsCpu":1,"limitsMem":"2Gi","name":"metrics-server","requestsCpu":1,"requestsMem":"2Gi"}]}}}]);
    AutopilotClusterMetadata metadatabody = new AutopilotClusterMetadata();
    metadatabody.setName("test-cluster-autopilot")
        .withAnnotations(listMetadataAnnotations);
    body.withSpec(specbody);
    body.withMetadata(metadatabody);
    body.withApiVersion("v3");
    body.withKind("Cluster");
    request.withBody(body);
    try {
        CreateAutopilotClusterResponse response = client.createAutopilotCluster(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 Autopilot cluster of v1.28 billed on a pay-per-use basis.

```
# coding: utf-8

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["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 = CreateAutopilotClusterRequest()
        extendParamSpec = AutopilotClusterExtendParam(
            enterprise_project_id="0"
        )
        serviceNetworkSpec = AutopilotServiceNetwork(
            i_pv4_cidr="10.247.0.0/16"
        )
        listSubnetsEniNetwork = [
```

```

AutopilotNetworkSubnet(
    subnet_id="186f9322-50c5-4e5a-91e3-47da86959afc"
)
]
eniNetworkSpec = AutopilotEniNetwork(
    subnets=listSubnetsEniNetwork
)
containerNetworkSpec = AutopilotContainerNetwork(
    mode="eni"
)
hostNetworkSpec = AutopilotHostNetwork(
    vpc="c6549063-d459-4ae1-9550-b5fec6741b0f",
    subnet="xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
)
specbody = AutopilotClusterSpec(
    category="Turbo",
    type="VirtualMachine",
    flavor="cce.autopilot.cluster",
    version="v1.28",
    description="",
    enable_snat=True,
    enable_swr_image_access=True,
    host_network=hostNetworkSpec,
    container_network=containerNetworkSpec,
    eni_network=eniNetworkSpec,
    service_network=serviceNetworkSpec,
    billing_mode=0,
    extend_param=extendParamSpec
)
listAnnotationsMetadata = {
    "cluster.install.addons/install": [{"addonTemplateName":"coredns","values":{"flavor":{"category":
["Autopilot"],"is_default":true,"name":"autopilot-flavor1","replicas":2,"resources":
[{"limitsCpu":1,"limitsMem":"2Gi","name":"coredns","requestsCpu":1,"requestsMem":"2Gi"}]}},
{"addonTemplateName":"metrics-server","values":{"flavor":{"category":["Autopilot"],"description":"custom
resources in autopilot cluster","is_default":true,"name":"autopilot-flavor1","replicas":2,"resources":
[{"limitsCpu":1,"limitsMem":"2Gi","name":"metrics-server","requestsCpu":1,"requestsMem":"2Gi"}]}]}]}
}
metadatabody = AutopilotClusterMetadata(
    name="test-cluster-autopilot",
    annotations=listAnnotationsMetadata
)
request.body = AutopilotCluster(
    spec=specbody,
    metadata=metadatabody,
    api_version="v3",
    kind="Cluster"
)
response = client.create_autopilot_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 Autopilot cluster of v1.28 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.CreateAutopilotClusterRequest{
        enterpriseProjectIdExtendParam:= "0"
        extendParamSpec := &model.AutopilotClusterExtendParam{
            EnterpriseProjectId: &enterpriseProjectIdExtendParam,
        }
        ipv4CIDRServiceNetwork:= "10.247.0.0/16"
        serviceNetworkSpec := &model.AutopilotServiceNetwork{
            IPv4CIDR: &ipv4CIDRServiceNetwork,
        }
        var listSubnetsEniNetwork = []model.AutopilotNetworkSubnet{
            {
                SubnetID: "186f9322-50c5-4e5a-91e3-47da86959afc",
            },
        }
        eniNetworkSpec := &model.AutopilotEniNetwork{
            Subnets: listSubnetsEniNetwork,
        }
        containerNetworkSpec := &model.AutopilotContainerNetwork{
            Mode: model.GetAutopilotContainerNetworkModeEnum().ENI,
        }
        hostNetworkSpec := &model.AutopilotHostNetwork{
            Vpc: "c6549063-d459-4ae1-9550-b5fec6741b0f",
            Subnet: "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
        }
        categorySpec:= model.GetAutopilotClusterSpecCategoryEnum().TURBO
        typeSpec:= model.GetAutopilotClusterSpecTypeEnum().VIRTUAL_MACHINE
        versionSpec:= "v1.28"
        descriptionSpec:= ""
        enableSnatSpec:= true
        enableSWRIImageAccessSpec:= true
        billingModeSpec:= int32(0)
        specbody := &model.AutopilotClusterSpec{
            Category: &categorySpec,
            Type: &typeSpec,
            Flavor: "cce.autopilot.cluster",
            Version: &versionSpec,
            Description: &descriptionSpec,
            EnableSnat: &enableSnatSpec,
            EnableSWRIImageAccess: &enableSWRIImageAccessSpec,
            HostNetwork: hostNetworkSpec,
            ContainerNetwork: containerNetworkSpec,
            EniNetwork: eniNetworkSpec,
            ServiceNetwork: serviceNetworkSpec,
            BillingMode: &billingModeSpec,
            ExtendParam: extendParamSpec,
        }
        var listAnnotationsMetadata = map[string]string{
```

```

"cluster.install.addons/install": "[{"addonTemplateName":"coredns","values":{"flavor":{"category":["Autopilot"],"is_default":true,"name":"autopilot-flavor1","replicas":2,"resources":[{"limitsCpu":1,"limitsMem":"2Gi","name":"coredns","requestsCpu":1,"requestsMem":"2Gi"}]}}, {"addonTemplateName":"metrics-server","values":{"flavor":{"category":["Autopilot"],"description":"custom resources in autopilot cluster","is_default":true,"name":"autopilot-flavor1","replicas":2,"resources":[{"limitsCpu":1,"limitsMem":"2Gi","name":"metrics-server","requestsCpu":1,"requestsMem":"2Gi"}]}]}],
}
metadatabody := &model.AutopilotClusterMetadata{
    Name: "test-cluster-autopilot",
    Annotations: listAnnotationsMetadata,
}
request.Body = &model.AutopilotCluster{
    Spec: specbody,
    Metadata: metadatabody,
    ApiVersion: "v3",
    Kind: "Cluster",
}
response, err := client.CreateAutopilotCluster(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 delivered. |

Error Codes

See [Error Codes](#).

4.1.2 Obtaining a Cluster

Function

This API is used to obtain details about a cluster.

NOTE

The URL for cluster management is in the format of <https://Endpoint/uri>, where **uri** specifies the resource path for API access.

Calling Method

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

URI

GET /autopilot/v3/projects/{project_id}/clusters/{cluster_id}

Table 4-33 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| project_id | Yes | String | <p>Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Project IDs of the account</p> <p>Default value: N/A</p> |
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |

Table 4-34 Query Parameters

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|--|
| detail | No | String | <p>Obtains details about a cluster.</p> <p>If this parameter is set to true, the add-on name (addonTemplateName), version (version), and status (status) from the list of installed add-ons in the cluster are obtained and added to the annotation.</p> |

Request Parameters

Table 4-35 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Response Parameters

Status code: 200

Table 4-36 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 | AutopilotCluster-Metadata object | Basic information about a cluster. Metadata is a collection of attributes. |
| spec | AutopilotCluster-Spec object | Element type of the collection class. The main part of a cluster object to be managed is in spec . CCE creates or updates objects by defining or updating spec . |
| status | AutopilotCluster-Status object | Cluster status |

Table 4-37 AutopilotClusterMetadata

| 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 resource. The value is automatically generated after the object is created. A custom value will not 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 is left blank, the alias of the cluster is the same as the cluster name. In the response body for creating a cluster, if the cluster alias is not configured, the cluster alias will not 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 are not used to 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. |
| 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 a cluster during the upgrade. Customized values are invalid.</p> |
| creationTimestamp | String | Time when a cluster was created |
| updateTimestamp | String | Time when a cluster was updated |

Table 4-38 AutopilotClusterSpec

| Parameter | Type | Description |
|-----------|--------|--|
| category | String | Cluster type. Only Turbo is supported. |
| type | String | Master node architecture. <ul style="list-style-type: none"> • VirtualMachine: The master node is an x86 server. |
| flavor | String | Cluster specifications: cce.autopilot.cluster |
| version | String | <p>Version of a cluster, which mirrors the baseline version of the Kubernetes community. The latest version is recommended.</p> <p>You can create clusters of the latest three versions on the CCE console. To learn which cluster versions are available, log in to the CCE console, create a cluster, and check supported cluster versions.</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. |

| Parameter | Type | Description |
|-----------------|------------------|--|
| platformVersion | String | <p>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. This parameter cannot be customized, and when you create a cluster, the latest corresponding platform version is selected automatically.</p> <p>The format of platformVersion is cce.X.Y.</p> <ul style="list-style-type: none"> • X: internal feature version, which indicates 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 released. No other modification is involved. The value starts from 0 and increases monotonically. |
| description | String | <p>Cluster description, for example, which purpose the cluster is intended to serve. By default, this field is left blank. To modify cluster description after a cluster is created, call the API for updating a specified cluster or go to the cluster details page on the CCE console. Only UTF-8 encoding is supported.</p> |
| customSan | Array of strings | <p>The custom SAN field in the API server certificate of a cluster. It complies with the format specifications defined by the SSL standard X509. It is not supported by Autopilot clusters.</p> <ol style="list-style-type: none"> 1. Duplicate names are not allowed. 2. The format 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> |

| Parameter | Type | Description |
|-----------------------|--|---|
| enableSnat | Boolean | Whether SNAT is configured for a cluster. This parameter is used and returned only by the Autopilot cluster creation API. After this function is enabled, the cluster can access the Internet through a NAT gateway. By default, the existing NAT gateway in the selected VPC is used. Otherwise, the system automatically creates a NAT gateway of the default specifications, binds an EIP to the NAT gateway, and configures SNAT rules. |
| enableSWRImage Access | Boolean | Whether a cluster is interconnected with SWR. This parameter is used and returned only by the Autopilot cluster creation API. To ensure that your cluster nodes can pull images from SWR, the existing SWR and OBS endpoints in the selected VPC are used by default. If not, new SWR and OBS endpoints will be automatically created. |
| enableAutopilot | Boolean | Whether the cluster is an Autopilot cluster |
| ipv6enable | Boolean | Whether a cluster uses the IPv6 mode. This parameter is not supported by Autopilot clusters. |
| hostNetwork | AutopilotHostNetwork object | Node networking parameters, including VPC and subnet ID. This field is mandatory because nodes in a cluster communicate with each other using a VPC. |
| containerNetwork | AutopilotContainerNetwork object | Container networking parameters, including the container network model and container CIDR block |
| eniNetwork | AutopilotEniNetwork object | Cloud Native 2.0 network settings |
| serviceNetwork | AutopilotServiceNetwork object | Service CIDR block, including IPv4 CIDR block |
| authentication | AutopilotAuthentication object | Configuration of the cluster authentication mode, which is not supported by Autopilot clusters |

| Parameter | Type | Description |
|------------------------|---|---|
| billingMode | Integer | Cluster billing mode <ul style="list-style-type: none"> • 0: pay-per-use This parameter defaults to pay-per-use . |
| kubernetesSvcIpRange | String | Service CIDR block or the IP address range which the kubernetes clusterIP must fall within. If this parameter is not specified during cluster creation, the default value 10.247.0.0/16 is used. This parameter is being deprecated, so you can use serviceNetwork instead. The new field contains the IPv4 CIDR blocks. |
| clusterTags | Array of AutopilotResourceTag objects | Cluster resource tags |
| kubeProxyMode | String | Service forwarding mode. <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. NOTE iptables is used by default. |
| az | String | AZ. This field is returned only for a query. For details about AZs supported by CCE, see Regions and Endpoints . |
| extendParam | AutopilotClusterExtendParam object | Extended field to decide whether a cluster will span across AZs or belong to a specified enterprise project, or whether a dedicated CCE cluster is to be created |
| configurationsOverride | Array of AutopilotPackageConfiguration objects | Overrides the default component configurations in a cluster. It is not supported in CCE Autopilot clusters. |

Table 4-39 AutopilotHostNetwork

| Parameter | Type | Description |
|-----------|--------|---|
| vpc | String | <p>ID of the VPC used to create a master node.</p> <p>Obtain the value using either of the following methods:</p> <ul style="list-style-type: none"> Method 1: Log in to the VPC console and view the VPC ID on the VPC details page. Method 2: Use the VPC API. For details, see Querying VPCs. |
| subnet | String | <p>Network ID of the subnet used to create a master node. Obtain the value using either of the following methods:</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. For details, see Querying Subnets. |

Table 4-40 AutopilotContainerNetwork

| Parameter | Type | Description |
|-----------|--------|---|
| mode | String | <p>Container network type</p> <ul style="list-style-type: none"> eni*: specifies the Cloud Native 2.0 network model. This model has integrated cloud native elastic network interfaces (ENIs), uses VPC CIDR blocks to allocate container IP addresses, and supports ELB passthrough networking for high network performance. Use this model when creating a cluster. |

Table 4-41 AutopilotEniNetwork

| Parameter | Type | Description |
|-----------|---|--|
| subnets | Array of AutopilotNetworkSubnet objects | IPv4 subnet ID list of the subnet where an ENI resides. Obtain the value using either of the following methods: <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 4-42 AutopilotNetworkSubnet

| Parameter | Type | Description |
|-----------|--------|--|
| subnetID | String | Specifies the IPv4 subnet ID of the subnet used to create control nodes and containers. (IPv6 is not supported.) To obtain the subnet ID, use either of the following methods: <ul style="list-style-type: none"> Method 1: Log in to the VPC console and click a subnet in the VPC. On the subnet details page, search for the IPv4 subnet ID. Method 2: Use the VPC API for obtaining subnets. For details, see Querying Subnets . |

Table 4-43 AutopilotServiceNetwork

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

Table 4-44 AutopilotAuthentication

| Parameter | Type | Description |
|-----------|--------|---|
| mode | String | Cluster authentication mode. The default value is rbac . |

Table 4-45 AutopilotResourceTag

| 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 4-46 AutopilotClusterExtendParam

| Parameter | Type | Description |
|---------------------|--------|--|
| enterpriseProjectId | String | ID of the enterprise project that a cluster belongs to NOTE <ul style="list-style-type: none"> An enterprise project can be configured only after the enterprise project function is enabled. |
| upgradeFrom | String | Records of how a cluster is upgraded to its current version |

Table 4-47 AutopilotPackageConfiguration

| Parameter | Type | Description |
|-----------|--------|----------------|
| name | String | Component name |

| Parameter | Type | Description |
|----------------|--|-------------------------------|
| configurations | Array of AutopilotConfigurationItem objects | Component configuration items |

Table 4-48 AutopilotConfigurationItem

| Parameter | Type | Description |
|-----------|--------|------------------------------------|
| name | String | Component configuration item name |
| value | Object | Component configuration item value |

Table 4-49 AutopilotClusterStatus

| 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 and needs to be manually deleted. ● 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. ● ResizeFailed: Cluster specification changing is abnormal. ● 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. ● Freezing: The cluster is being frozen. ● Frozen: The cluster has been frozen. ● UnFreezing: The cluster is being unfrozen. ● 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. You can manually delete the cluster. |

| Parameter | Type | Description |
|--------------|---|--|
| jobID | String | <p>ID of the task associated with the current cluster status. 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 being 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 | Why a cluster changes to the current state. This parameter is returned if the cluster is not in the Available state. |
| message | String | Detailed information about why a cluster changes to the current state. This parameter is returned if the cluster is not in the Available state. |
| endpoints | Array of AutopilotCluster-Endpoints objects | Access address of kube-apiserver in a cluster |
| isLocked | Boolean | CBC resource locked |
| lockScene | String | Scenario where the CBC resource is locked |
| lockSource | String | Resource locked |
| lockSourceId | String | ID of a locked resource |
| deleteOption | Object | Status of the configuration to be deleted. This parameter is contained only in the response to the deletion request. |
| deleteStatus | Object | Information of the status to be deleted. This parameter is contained only in the response to the deletion request. |

Table 4-50 AutopilotClusterEndpoints

| Parameter | Type | Description |
|-----------|--------|---|
| url | String | Access address of kube-apiserver in a 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 cluster is obtained.

```
{
  "kind": "Cluster",
  "apiVersion": "v3",
  "metadata": {
    "name": "autopilot-test-v1274",
    "uid": "087dc72a-1ff6-11ef-af74-0255ac10010b",
    "alias": "autopilot-test-v1274",
    "annotations": {
      "enableAutopilot": "true"
    }
  },
  "labels": {
    "FeatureGates": "arpOptimization,elbv3,xGPU"
  },
  "creationTimestamp": "2024-06-01 09:05:03.665134 +0000 UTC",
  "updateTimestamp": "2024-06-04 12:54:27.803532 +0000 UTC"
},
"spec": {
  "category": "Turbo",
  "type": "VirtualMachine",
  "flavor": "cce.autopilot.cluster",
  "version": "v1.27",
  "platformVersion": "cce.6.0",
  "hostNetwork": {
    "vpc": "13cd773f-7f9f-4821-b9b8-dba5b351e1ec",
    "subnet": "19e3960d-b9ae-4d2d-b3a4-92ff56ae0301"
  },
  "containerNetwork": {
    "mode": "eni"
  },
  "eniNetwork": {
    "subnets": [ {
      "subnetID": "f061c486-a190-4e2e-993c-5bc3dc9d65a8"
    } ]
  },
  "serviceNetwork": {
    "IPv4CIDR": "10.247.0.0/16"
  },
  "authentication": {
    "mode": "rbac"
  }
}
```

```
    },
    "billingMode" : 0,
    "kubernetesSvcIpRange" : "10.247.0.0/16",
    "kubeProxyMode" : "iptables",
    "az" : "cn-north-7c",
    "extendParam" : {
      "enterpriseProjectId" : "0",
      "upgradefrom" : ""
    }
  },
  "status" : {
    "phase" : "Available",
    "endpoints" : [ {
      "url" : "https://087dc72a-1ff6-11ef-af74-0255ac10010b.cluster.cce.cn-north-7.myhuaweicloud.com: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 ShowAutopilotClusterSolution {

    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();
        ShowAutopilotClusterRequest request = new ShowAutopilotClusterRequest();
        request.withClusterId("{cluster_id}");
        try {
            ShowAutopilotClusterResponse response = client.showAutopilotCluster(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

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["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 = ShowAutopilotClusterRequest()
        request.cluster_id = "{cluster_id}"
        response = client.show_autopilot_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.ShowAutopilotClusterRequest{}
request.ClusterId = "{cluster_id}"
response, err := client.ShowAutopilotCluster(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 cluster is obtained. |

Error Codes

See [Error Codes](#).

4.1.3 Obtaining Clusters in a Project

Function

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

Calling Method

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

URI

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

Table 4-51 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| project_id | Yes | String | <p>Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Project IDs of the account</p> <p>Default value: N/A</p> |

Table 4-52 Query Parameters

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|--|
| detail | No | String | <p>Obtains details about a cluster.</p> <p>If this parameter is set to true, the add-on name (addonTemplateName), version (version), and status (status) from the list of installed add-ons in the cluster are obtained and added to the annotation.</p> |

| 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 and needs to be manually deleted. • 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. • ResizeFailed: Cluster specification changing is abnormal. • 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. • Freezing: The cluster is being frozen. • Frozen: The cluster has been frozen. • UnFreezing: The cluster is being unfrozen. • Awaking: The cluster is being woken up from hibernation. • Empty: The cluster does not have any resources. This field is discarded. |

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|---|
| | | | <ul style="list-style-type: none"> • Error: Resources in the cluster are abnormal. You can manually delete the cluster. |
| type | No | String | Cluster type. Options: <ul style="list-style-type: none"> • VirtualMachine: CCE cluster |
| version | No | String | Obtains the version of a cluster. |

Request Parameters

Table 4-53 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Response Parameters

Status code: 200

Table 4-54 Response body parameters

| Parameter | Type | Description |
|------------|---|---|
| kind | String | Api type |
| apiVersion | String | API version |
| items | Array of AutopilotCluster objects | A list of details for all clusters in the current project. You can find the corresponding clusters based on the value of <code>items.metadata.name</code> . |

Table 4-55 AutopilotCluster

| 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 | AutopilotCluster-Metadata object | Basic information about a cluster. Metadata is a collection of attributes. |

| Parameter | Type | Description |
|-----------|--------------------------------------|--|
| spec | AutopilotClusterSpec object | Element type of the collection class. The main part of a cluster object to be managed is in spec . CCE creates or updates objects by defining or updating spec . |
| status | AutopilotClusterStatus object | Element type of the collection class, which is used to record the statuses of an object in the system, including the cluster status and the ID of the cluster creation job |

Table 4-56 AutopilotClusterMetadata

| 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 resource. The value is automatically generated after the object is created. A custom value will not 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 is left blank, the alias of the cluster is the same as the cluster name. In the response body for creating a cluster, if the cluster alias is not configured, the cluster alias will not be returned. |

| Parameter | Type | Description |
|-------------------|---------------------|---|
| 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 are not used to 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. |
| labels | Map<String,String > | Cluster labels in the format of key-value pairs. <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 a cluster during the upgrade. Customized values are invalid.</p> |
| creationTimestamp | String | Time when a cluster was created |
| updateTimestamp | String | Time when a cluster was updated |

Table 4-57 AutopilotClusterSpec

| Parameter | Type | Description |
|-----------|--------|--|
| category | String | Cluster type. Only Turbo is supported. |
| type | String | Master node architecture. <ul style="list-style-type: none"> VirtualMachine: The master node is an x86 server. |
| flavor | String | Cluster specifications: cce.autopilot.cluster |

| Parameter | Type | Description |
|-----------------|--------|--|
| version | String | <p>Version of a cluster, which mirrors the baseline version of the Kubernetes community. The latest version is recommended.</p> <p>You can create clusters of the latest three versions on the CCE console. To learn which cluster versions are available, log in to the CCE console, create a cluster, and check supported cluster versions.</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. |
| platformVersion | String | <p>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. This parameter cannot be customized, and when you create a cluster, the latest corresponding platform version is selected automatically.</p> <p>The format of platformVersion is cce.X.Y.</p> <ul style="list-style-type: none"> X: internal feature version, which indicates 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 released. 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 blank. To modify cluster description after a cluster is created, call the API for updating a specified cluster or go to the cluster details page on the CCE console. Only UTF-8 encoding is supported. |
| customSan | Array of strings | The custom SAN field in the API server certificate of a cluster. It complies with the format specifications defined by the SSL standard X509. It is not supported by Autopilot clusters. <ul style="list-style-type: none"> 1. Duplicate names are not allowed. 2. The format 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> |
| enableSnat | Boolean | Whether SNAT is configured for a cluster. This parameter is used and returned only by the Autopilot cluster creation API. After this function is enabled, the cluster can access the Internet through a NAT gateway. By default, the existing NAT gateway in the selected VPC is used. Otherwise, the system automatically creates a NAT gateway of the default specifications, binds an EIP to the NAT gateway, and configures SNAT rules. |
| enableSWRImage Access | Boolean | Whether a cluster is interconnected with SWR. This parameter is used and returned only by the Autopilot cluster creation API. To ensure that your cluster nodes can pull images from SWR, the existing SWR and OBS endpoints in the selected VPC are used by default. If not, new SWR and OBS endpoints will be automatically created. |
| enableAutopilot | Boolean | Whether the cluster is an Autopilot cluster |

| Parameter | Type | Description |
|----------------------|---|---|
| ipv6enable | Boolean | Whether a cluster uses the IPv6 mode. This parameter is not supported by Autopilot clusters. |
| hostNetwork | AutopilotHostNetwork object | Node networking parameters, including VPC and subnet ID. This field is mandatory because nodes in a cluster communicate with each other using a VPC. |
| containerNetwork | AutopilotContainerNetwork object | Container networking parameters, including the container network model and container CIDR block |
| eniNetwork | AutopilotEniNetwork object | Cloud Native 2.0 network settings |
| serviceNetwork | AutopilotServiceNetwork object | Service CIDR block, including IPv4 CIDR block |
| authentication | AutopilotAuthentication object | Configuration of the cluster authentication mode, which is not supported by Autopilot clusters |
| billingMode | Integer | Cluster billing mode <ul style="list-style-type: none"> • 0: pay-per-use This parameter defaults to pay-per-use . |
| kubernetesSvcIpRange | String | Service CIDR block or the IP address range which the kubernetes clusterIP must fall within. If this parameter is not specified during cluster creation, the default value 10.247.0.0/16 is used. This parameter is being deprecated, so you can use serviceNetwork instead. The new field contains the IPv4 CIDR blocks. |
| clusterTags | Array of AutopilotResourceTag objects | Cluster resource tags |

| Parameter | Type | Description |
|-----------------------------|--|---|
| kubeProxyMode | String | Service forwarding mode. <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. <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 | AutopilotCluster-ExtendParam object | Extended field to decide whether a cluster will span across AZs or belong to a specified enterprise project, or whether a dedicated CCE cluster is to be created |
| configurationsO- verride | Array of AutopilotPackageConfiguration objects | Overrides the default component configurations in a cluster. It is not supported in CCE Autopilot clusters. |

Table 4-58 AutopilotHostNetwork

| Parameter | Type | Description |
|-----------|--------|---|
| vpc | String | ID of the VPC used to create a master node. Obtain the value using either of the following methods: <ul style="list-style-type: none"> Method 1: Log in to the VPC console and view the VPC ID on the VPC details page. Method 2: Use the VPC API. For details, see Querying VPCs. |

| Parameter | Type | Description |
|-----------|--------|--|
| subnet | String | <p>Network ID of the subnet used to create a master node. Obtain the value using either of the following methods:</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> |

Table 4-59 AutopilotContainerNetwork

| Parameter | Type | Description |
|-----------|--------|---|
| mode | String | <p>Container network type</p> <ul style="list-style-type: none"> eni*: specifies the Cloud Native 2.0 network model. This model has integrated cloud native elastic network interfaces (ENIs), uses VPC CIDR blocks to allocate container IP addresses, and supports ELB passthrough networking for high network performance. Use this model when creating a cluster. |

Table 4-60 AutopilotEniNetwork

| Parameter | Type | Description |
|-----------|---|---|
| subnets | Array of AutopilotNetworkSubnet objects | <p>IPv4 subnet ID list of the subnet where an ENI resides. Obtain the value using either of the following methods:</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 4-61 AutopilotNetworkSubnet

| Parameter | Type | Description |
|-----------|--------|--|
| subnetID | String | Specifies the IPv4 subnet ID of the subnet used to create control nodes and containers. (IPv6 is not supported.) To obtain the subnet ID, use either of the following methods: <ul style="list-style-type: none"> Method 1: Log in to the VPC console and click a subnet in the VPC. On the subnet details page, search for the IPv4 subnet ID. Method 2: Use the VPC API for obtaining subnets. For details, see Querying Subnets . |

Table 4-62 AutopilotServiceNetwork

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

Table 4-63 AutopilotAuthentication

| Parameter | Type | Description |
|-----------|--------|---|
| mode | String | Cluster authentication mode. The default value is rbac . |

Table 4-64 AutopilotResourceTag

| 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>. |

| Parameter | Type | Description |
|-----------|--------|--|
| 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 4-65 AutopilotClusterExtendParam

| Parameter | Type | Description |
|---------------------|--------|--|
| enterpriseProjectId | String | ID of the enterprise project that a cluster belongs to NOTE <ul style="list-style-type: none"> • An enterprise project can be configured only after the enterprise project function is enabled. |
| upgradeFrom | String | Records of how a cluster is upgraded to its current version |

Table 4-66 AutopilotPackageConfiguration

| Parameter | Type | Description |
|----------------|---|-------------------------------|
| name | String | Component name |
| configurations | Array of AutopilotConfigurationItem objects | Component configuration items |

Table 4-67 AutopilotConfigurationItem

| Parameter | Type | Description |
|-----------|--------|------------------------------------|
| name | String | Component configuration item name |
| value | Object | Component configuration item value |

Table 4-68 AutopilotClusterStatus

| 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 and needs to be manually deleted. ● 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. ● ResizeFailed: Cluster specification changing is abnormal. ● 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. ● Freezing: The cluster is being frozen. ● Frozen: The cluster has been frozen. ● UnFreezing: The cluster is being unfrozen. ● 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. You can manually delete the cluster. |

| Parameter | Type | Description |
|--------------|---|--|
| jobID | String | <p>ID of the task associated with the current cluster status. 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 being 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 | Why a cluster changes to the current state. This parameter is returned if the cluster is not in the Available state. |
| message | String | Detailed information about why a cluster changes to the current state. This parameter is returned if the cluster is not in the Available state. |
| endpoints | Array of AutopilotCluster-Endpoints objects | Access address of kube-apiserver in a cluster |
| isLocked | Boolean | CBC resource locked |
| lockScene | String | Scenario where the CBC resource is locked |
| lockSource | String | Resource locked |
| lockSourceId | String | ID of a locked resource |
| deleteOption | Object | Status of the configuration to be deleted. This parameter is contained only in the response to the deletion request. |
| deleteStatus | Object | Information of the status to be deleted. This parameter is contained only in the response to the deletion request. |

Table 4-69 AutopilotClusterEndpoints

| Parameter | Type | Description |
|-----------|--------|---|
| url | String | Access address of kube-apiserver in a 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 obtained.

```
{
  "kind": "Cluster",
  "apiVersion": "v3",
  "items": [ {
    "kind": "Cluster",
    "apiVersion": "v3",
    "metadata": {
      "name": "test",
      "uid": "de63df79-1de1-11ef-95a4-0255ac1007fa",
      "creationTimestamp": "2024-05-29 17:35:40.770923 +0000 UTC",
      "updateTimestamp": "2024-06-04 08:14:13.881756 +0000 UTC",
      "labels": {
        "FeatureGates": "arpOptimization,elbv3,xGPU"
      }
    },
    "alias": "s00648239-b003-v127"
  } ],
  "spec": {
    "category": "Turbo",
    "type": "VirtualMachine",
    "flavor": "cce.autopilot.cluster",
    "version": "v1.27",
    "hostNetwork": {
      "vpc": "26958bf6-9ce3-4184-9e19-d793880a162b",
      "subnet": "df93b82f-5196-43a9-a3f1-78deeab504eb"
    },
    "containerNetwork": {
      "mode": "eni"
    },
    "eniNetwork": {
      "subnets": [ {
        "subnetID": "97205694-3537-45b6-9459-c98e9704574a"
      } ]
    },
    "serviceNetwork": {
      "IPv4CIDR": "10.247.0.0/16"
    },
    "authentication": {
      "mode": "rbac"
    }
  }
}
```

```
"billingMode" : 0,
"kubernetesSvcIpRange" : "10.247.0.0/16",
"kubeProxyMode" : "iptables",
"az" : "cn-north-7c",
"extendParam" : {
  "enterpriseProjectId" : "0",
  "upgradefrom" : ""
}
},
"status" : {
  "phase" : "Available",
  "endpoints" : [ {
    "url" : "https://de63df79-1de1-11ef-95a4-0255ac1007fa.cluster.cce.cn-
north-7.myhuaweicloud.com: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 ListAutopilotClustersSolution {

    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();
        ListAutopilotClustersRequest request = new ListAutopilotClustersRequest();
        try {
            ListAutopilotClustersResponse response = client.listAutopilotClusters(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

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["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 = ListAutopilotClustersRequest()
        response = client.list_autopilot_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.ListAutopilotClustersRequest{}
response, err := client.ListAutopilotClusters(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 obtained. |

Error Codes

See [Error Codes](#).

4.1.4 Updating a Cluster

Function

This API is used to update a cluster.

NOTE

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

Calling Method

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

URI

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

Table 4-70 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| project_id | Yes | String | <p>Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Project IDs of the account</p> <p>Default value: N/A</p> |
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |

Request Parameters

Table 4-71 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Table 4-72 Request body parameters

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--|-----------------------------|
| spec | Yes | AutopilotClusterInformationSpec object | Detailed cluster parameters |

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--|--|
| metadata | No | AutopilotClusterMetadataForUpdate object | Basic cluster information, including name-related fields |

Table 4-73 AutopilotClusterInformationSpec

| Parameter | Mandatory | Type | Description |
|-------------|-----------|--|---|
| description | No | String | <p>Description of a cluster.</p> <ul style="list-style-type: none"> The value can be 0 to 200 characters and cannot contain the following special characters: ~\$%^&*<>[]{}()'"#\ Only description of available clusters can be modified. |
| customSan | No | Array of strings | <p>The custom SAN field in the API server certificate of a cluster. It complies with the format specifications defined by the SSL standard X509. It is not supported by Autopilot clusters.</p> <ol style="list-style-type: none"> Duplicate names are not allowed. The format 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> |
| eniNetwork | No | AutopilotEniNetworkUpdate object | Cloud native 2.0 network settings, including the container subnet information of a cluster |

Table 4-74 AutopilotEniNetworkUpdate

| Parameter | Mandatory | Type | Description |
|-----------|-----------|---|--|
| subnets | No | Array of AutopilotNetworkSubnet objects | List of IPv4 subnet IDs Subnets can be added and existing subnets can be deleted. |

Table 4-75 AutopilotNetworkSubnet

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|--|
| subnetID | Yes | String | Specifies the IPv4 subnet ID of the subnet used to create control nodes and containers. (IPv6 is not supported.) To obtain the subnet ID, use either of the following methods: <ul style="list-style-type: none"> • Method 1: Log in to the VPC console and click a subnet in the VPC. On the subnet details page, search for the IPv4 subnet ID. • Method 2: Use the VPC API for obtaining subnets. For details, see Querying Subnets . |

Table 4-76 AutopilotClusterMetadataForUpdate

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|--|
| alias | No | String | Alias of a 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. A cluster alias must be unique. If this parameter is left blank, the alias is not changed. |

Response Parameters

Status code: 200

Table 4-77 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 | AutopilotCluster-Metadata object | Basic information about a cluster. Metadata is a collection of attributes. |
| spec | AutopilotCluster-Spec object | Element type of the collection class. The main part of a cluster object to be managed is in spec . CCE creates or updates objects by defining or updating spec . |
| status | AutopilotCluster-Status object | Element type of the collection class, which is used to record the statuses of an object in the system, including the cluster status and the ID of the cluster creation job |

Table 4-78 AutopilotClusterMetadata

| 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 resource. The value is automatically generated after the object is created. A custom value will not 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 is left blank, the alias of the cluster is the same as the cluster name. In the response body for creating a cluster, if the cluster alias is not configured, the cluster alias will not 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 are not used to 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. |
| 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 a cluster during the upgrade. Customized values are invalid.</p> |
| creationTimestamp | String | Time when a cluster was created |
| updateTimestamp | String | Time when a cluster was updated |

Table 4-79 AutopilotClusterSpec

| Parameter | Type | Description |
|-----------|--------|--|
| category | String | Cluster type. Only Turbo is supported. |
| type | String | Master node architecture. <ul style="list-style-type: none"> • VirtualMachine: The master node is an x86 server. |
| flavor | String | Cluster specifications: cce.autopilot.cluster |
| version | String | <p>Version of a cluster, which mirrors the baseline version of the Kubernetes community. The latest version is recommended.</p> <p>You can create clusters of the latest three versions on the CCE console. To learn which cluster versions are available, log in to the CCE console, create a cluster, and check supported cluster versions.</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. |

| Parameter | Type | Description |
|-----------------|------------------|--|
| platformVersion | String | <p>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. This parameter cannot be customized, and when you create a cluster, the latest corresponding platform version is selected automatically.</p> <p>The format of platformVersion is cce.X.Y.</p> <ul style="list-style-type: none"> • X: internal feature version, which indicates 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 released. No other modification is involved. The value starts from 0 and increases monotonically. |
| description | String | <p>Cluster description, for example, which purpose the cluster is intended to serve. By default, this field is left blank. To modify cluster description after a cluster is created, call the API for updating a specified cluster or go to the cluster details page on the CCE console. Only UTF-8 encoding is supported.</p> |
| customSan | Array of strings | <p>The custom SAN field in the API server certificate of a cluster. It complies with the format specifications defined by the SSL standard X509. It is not supported by Autopilot clusters.</p> <ol style="list-style-type: none"> 1. Duplicate names are not allowed. 2. The format 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> |

| Parameter | Type | Description |
|-----------------------|--|---|
| enableSnat | Boolean | Whether SNAT is configured for a cluster. This parameter is used and returned only by the Autopilot cluster creation API. After this function is enabled, the cluster can access the Internet through a NAT gateway. By default, the existing NAT gateway in the selected VPC is used. Otherwise, the system automatically creates a NAT gateway of the default specifications, binds an EIP to the NAT gateway, and configures SNAT rules. |
| enableSWRImage Access | Boolean | Whether a cluster is interconnected with SWR. This parameter is used and returned only by the Autopilot cluster creation API. To ensure that your cluster nodes can pull images from SWR, the existing SWR and OBS endpoints in the selected VPC are used by default. If not, new SWR and OBS endpoints will be automatically created. |
| enableAutopilot | Boolean | Whether the cluster is an Autopilot cluster |
| ipv6enable | Boolean | Whether a cluster uses the IPv6 mode. This parameter is not supported by Autopilot clusters. |
| hostNetwork | AutopilotHostNetwork object | Node networking parameters, including VPC and subnet ID. This field is mandatory because nodes in a cluster communicate with each other using a VPC. |
| containerNetwork | AutopilotContainerNetwork object | Container networking parameters, including the container network model and container CIDR block |
| eniNetwork | AutopilotEniNetwork object | Cloud Native 2.0 network settings |
| serviceNetwork | AutopilotServiceNetwork object | Service CIDR block, including IPv4 CIDR block |
| authentication | AutopilotAuthentication object | Configuration of the cluster authentication mode, which is not supported by Autopilot clusters |

| Parameter | Type | Description |
|------------------------|---|---|
| billingMode | Integer | Cluster billing mode <ul style="list-style-type: none"> • 0: pay-per-use This parameter defaults to pay-per-use . |
| kubernetesSvcIpRange | String | Service CIDR block or the IP address range which the kubernetes clusterIP must fall within. If this parameter is not specified during cluster creation, the default value 10.247.0.0/16 is used. This parameter is being deprecated, so you can use serviceNetwork instead. The new field contains the IPv4 CIDR blocks. |
| clusterTags | Array of AutopilotResourceTag objects | Cluster resource tags |
| kubeProxyMode | String | Service forwarding mode. <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. NOTE iptables is used by default. |
| az | String | AZ. This field is returned only for a query. For details about AZs supported by CCE, see Regions and Endpoints . |
| extendParam | AutopilotClusterExtendParam object | Extended field to decide whether a cluster will span across AZs or belong to a specified enterprise project, or whether a dedicated CCE cluster is to be created |
| configurationsOverride | Array of AutopilotPackageConfiguration objects | Overrides the default component configurations in a cluster. It is not supported in CCE Autopilot clusters. |

Table 4-80 AutopilotHostNetwork

| Parameter | Type | Description |
|-----------|--------|---|
| vpc | String | <p>ID of the VPC used to create a master node.</p> <p>Obtain the value using either of the following methods:</p> <ul style="list-style-type: none"> Method 1: Log in to the VPC console and view the VPC ID on the VPC details page. Method 2: Use the VPC API. For details, see Querying VPCs. |
| subnet | String | <p>Network ID of the subnet used to create a master node. Obtain the value using either of the following methods:</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. For details, see Querying Subnets. |

Table 4-81 AutopilotContainerNetwork

| Parameter | Type | Description |
|-----------|--------|---|
| mode | String | <p>Container network type</p> <ul style="list-style-type: none"> eni*: specifies the Cloud Native 2.0 network model. This model has integrated cloud native elastic network interfaces (ENIs), uses VPC CIDR blocks to allocate container IP addresses, and supports ELB passthrough networking for high network performance. Use this model when creating a cluster. |

Table 4-82 AutopilotEniNetwork

| Parameter | Type | Description |
|-----------|---|--|
| subnets | Array of AutopilotNetworkSubnet objects | IPv4 subnet ID list of the subnet where an ENI resides. Obtain the value using either of the following methods: <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 4-83 AutopilotNetworkSubnet

| Parameter | Type | Description |
|-----------|--------|--|
| subnetID | String | Specifies the IPv4 subnet ID of the subnet used to create control nodes and containers. (IPv6 is not supported.) To obtain the subnet ID, use either of the following methods: <ul style="list-style-type: none"> Method 1: Log in to the VPC console and click a subnet in the VPC. On the subnet details page, search for the IPv4 subnet ID. Method 2: Use the VPC API for obtaining subnets. For details, see Querying Subnets . |

Table 4-84 AutopilotServiceNetwork

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

Table 4-85 AutopilotAuthentication

| Parameter | Type | Description |
|-----------|--------|---|
| mode | String | Cluster authentication mode. The default value is rbac . |

Table 4-86 AutopilotResourceTag

| 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 4-87 AutopilotClusterExtendParam

| Parameter | Type | Description |
|---------------------|--------|--|
| enterpriseProjectId | String | ID of the enterprise project that a cluster belongs to NOTE <ul style="list-style-type: none"> An enterprise project can be configured only after the enterprise project function is enabled. |
| upgradeFrom | String | Records of how a cluster is upgraded to its current version |

Table 4-88 AutopilotPackageConfiguration

| Parameter | Type | Description |
|-----------|--------|----------------|
| name | String | Component name |

| Parameter | Type | Description |
|----------------|---|-------------------------------|
| configurations | Array of AutopilotConfigurationItem objects | Component configuration items |

Table 4-89 AutopilotConfigurationItem

| Parameter | Type | Description |
|-----------|--------|------------------------------------|
| name | String | Component configuration item name |
| value | Object | Component configuration item value |

Table 4-90 AutopilotClusterStatus

| 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 and needs to be manually deleted. ● 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. ● ResizeFailed: Cluster specification changing is abnormal. ● 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. ● Freezing: The cluster is being frozen. ● Frozen: The cluster has been frozen. ● UnFreezing: The cluster is being unfrozen. ● 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. You can manually delete the cluster. |

| Parameter | Type | Description |
|--------------|---|--|
| jobID | String | <p>ID of the task associated with the current cluster status. 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 being 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 | Why a cluster changes to the current state. This parameter is returned if the cluster is not in the Available state. |
| message | String | Detailed information about why a cluster changes to the current state. This parameter is returned if the cluster is not in the Available state. |
| endpoints | Array of AutopilotCluster-Endpoints objects | Access address of kube-apiserver in a cluster |
| isLocked | Boolean | CBC resource locked |
| lockScene | String | Scenario where the CBC resource is locked |
| lockSource | String | Resource locked |
| lockSourceId | String | ID of a locked resource |
| deleteOption | Object | Status of the configuration to be deleted. This parameter is contained only in the response to the deletion request. |
| deleteStatus | Object | Information of the status to be deleted. This parameter is contained only in the response to the deletion request. |

Table 4-91 AutopilotClusterEndpoints

| Parameter | Type | Description |
|-----------|--------|---|
| url | String | Access address of kube-apiserver in a 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

Update the description of a cluster.

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

Example Responses

Status code: 200

The cluster is updated.

```
{
  "kind": "Cluster",
  "apiVersion": "v3",
  "metadata": {
    "name": "s00842745-128-3-r0",
    "uid": "c82a7d44-1cc4-11ef-8460-0255ac101780",
    "creationTimestamp": "2024-05-28 07:34:56.917029 +0000 UTC",
    "updateTimestamp": "2024-06-04 13:12:35.749294 +0000 UTC",
    "labels": {
      "FeatureGates": "arpOptimization,elbv3,xGPU"
    },
    "annotations": {
      "enableAutopilot": "true",
      "feature:supportNodePoolScaleGroup": "true"
    },
    "alias": "s00842745-128-3-r0"
  },
  "spec": {
    "category": "Turbo",
    "type": "VirtualMachine",
    "flavor": "cce.autopilot.cluster",
    "version": "v1.28",
    "platformVersion": "cce.3.0",
    "description": "new description",
    "hostNetwork": {
      "vpc": "f9122377-7b2e-49c9-ab9e-bf0bfdd807b4",
      "subnet": "6b757878-c428-4e76-a7e9-5e3853778d5d"
    },
    "containerNetwork": {
      "mode": "eni"
    },
    "eniNetwork": {
      "subnets": [ {

```

```
"subnetID" : "b04a4b46-9f99-44a1-9a98-de52e549e68b"
  } ]
},
"serviceNetwork" : {
  "IPv4CIDR" : "10.247.0.0/16"
},
"authentication" : {
  "mode" : "rbac"
},
"billingMode" : 0,
"kubernetesSvcIcpRange" : "10.247.0.0/16",
"kubeProxyMode" : "iptables",
"az" : "cn-north-7c",
"extendParam" : {
  "enterpriseProjectId" : "5ebc44c1-617b-4d93-8d49-895b8a457a1f",
  "upgradefrom" : ""
}
},
"status" : {
  "phase" : "Available",
  "endpoints" : [ {
    "url" : "https://c82a7d44-1cc4-11ef-8460-0255ac101780.cluster.cce.cn-
north-7.myhuaweicloud.com:5443",
    "type" : "Internal"
  } ]
}
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Update 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 UpdateAutopilotClusterSolution {

    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();
UpdateAutopilotClusterRequest request = new UpdateAutopilotClusterRequest();
request.withClusterId("{cluster_id}");
AutopilotClusterInformation body = new AutopilotClusterInformation();
AutopilotClusterInformationSpec specbody = new AutopilotClusterInformationSpec();
specbody.withDescription("new description");
body.withSpec(specbody);
request.withBody(body);
try {
    UpdateAutopilotClusterResponse response = client.updateAutopilotCluster(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 description of a cluster.

```
# coding: utf-8

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["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 = UpdateAutopilotClusterRequest()
        request.cluster_id = "{cluster_id}"
        specbody = AutopilotClusterInformationSpec(
            description="new description"
        )
        request.body = AutopilotClusterInformation(
            spec=specbody
        )
        response = client.update_autopilot_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

Update 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.UpdateAutopilotClusterRequest{}
    request.ClusterId = "{cluster_id}"
    descriptionSpec := "new description"
    specbody := &model.AutopilotClusterInformationSpec{
        Description: &descriptionSpec,
    }
    request.Body = &model.AutopilotClusterInformation{
        Spec: specbody,
    }
    response, err := client.UpdateAutopilotCluster(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 is updated. |

Error Codes

See [Error Codes](#).

4.1.5 Deleting a Cluster

Function

This API is used to delete a cluster.

 **NOTE**

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

Calling Method

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

URI

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

Table 4-92 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| project_id | Yes | String | <p>Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Project IDs of the account</p> <p>Default value: N/A</p> |

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|--|
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |

Table 4-93 Query Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|--|
| delete_efs | No | String | <p>Whether to delete an SFS Turbo volume.</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.) |

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| delete_eni | No | String | <p>Whether to delete an ENI port (native ENI).</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. This is the default option. • 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. |
| delete_net | No | String | <p>Whether to delete cluster Service or ingress resources, such as a load balancer.</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. This is the default option. • 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. |

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|---|
| delete_obs | No | String | <p>Whether to delete an OBS volume.</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. |
| delete_sfs30 | No | String | <p>Whether to delete an SFS 3.0 volume.</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. |

| Parameter | Mandatory | Type | Description |
|--------------------|-----------|--------|--|
| lts_reclaim_policy | No | String | <p>Whether to delete an LTS resource, such as a log group or a log stream.</p> <p>Options:</p> <ul style="list-style-type: none"> • Delete_Log_Group: The system starts to delete a log group. If the deletion fails, no deletion retry is performed, and subsequent processes will proceed. • Delete_Master_Log_Stream: The system starts to delete a master log stream. If the deletion fails, no deletion retry is performed, and subsequent processes will proceed. This is the default option. • Retain: The deletion is skipped. |

Request Parameters

Table 4-94 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Response Parameters

Status code: 200

Table 4-95 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 | AutopilotCluster-Metadata object | Basic information about a cluster. Metadata is a collection of attributes. |
| spec | AutopilotCluster-Spec object | Element type of the collection class. The main part of a cluster object to be managed is in spec . CCE creates or updates objects by defining or updating spec . |
| status | AutopilotCluster-Status object | Element type of the collection class, which is used to record the statuses of an object in the system, including the cluster status and the ID of the cluster creation job |

Table 4-96 AutopilotClusterMetadata

| 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 resource. The value is automatically generated after the object is created. A custom value will not 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 is left blank, the alias of the cluster is the same as the cluster name. In the response body for creating a cluster, if the cluster alias is not configured, the cluster alias will not 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 are not used to 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. |

| Parameter | Type | Description |
|-------------------|--------------------|--|
| 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 a cluster during the upgrade. Customized values are invalid. |
| creationTimestamp | String | Time when a cluster was created |
| updateTimestamp | String | Time when a cluster was updated |

Table 4-97 AutopilotClusterSpec

| Parameter | Type | Description |
|-----------|--------|---|
| category | String | Cluster type. Only Turbo is supported. |
| type | String | Master node architecture. <ul style="list-style-type: none"> VirtualMachine: The master node is an x86 server. |
| flavor | String | Cluster specifications: cce.autopilot.cluster |
| version | String | Version of a cluster, which mirrors the baseline version of the Kubernetes community. The latest version is recommended. You can create clusters of the latest three versions on the CCE console. To learn which cluster versions are available, log in to the CCE console, create a cluster, and check supported cluster versions. 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. NOTE <ul style="list-style-type: none"> If not specified, a cluster of the latest version will be created. |

| Parameter | Type | Description |
|-----------------|------------------|--|
| platformVersion | String | <p>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. This parameter cannot be customized, and when you create a cluster, the latest corresponding platform version is selected automatically.</p> <p>The format of platformVersion is cce.X.Y.</p> <ul style="list-style-type: none"> • X: internal feature version, which indicates 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 released. No other modification is involved. The value starts from 0 and increases monotonically. |
| description | String | <p>Cluster description, for example, which purpose the cluster is intended to serve. By default, this field is left blank. To modify cluster description after a cluster is created, call the API for updating a specified cluster or go to the cluster details page on the CCE console. Only UTF-8 encoding is supported.</p> |
| customSan | Array of strings | <p>The custom SAN field in the API server certificate of a cluster. It complies with the format specifications defined by the SSL standard X509. It is not supported by Autopilot clusters.</p> <ol style="list-style-type: none"> 1. Duplicate names are not allowed. 2. The format 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> |

| Parameter | Type | Description |
|-----------------------|---|---|
| enableSnat | Boolean | Whether SNAT is configured for a cluster. This parameter is used and returned only by the Autopilot cluster creation API. After this function is enabled, the cluster can access the Internet through a NAT gateway. By default, the existing NAT gateway in the selected VPC is used. Otherwise, the system automatically creates a NAT gateway of the default specifications, binds an EIP to the NAT gateway, and configures SNAT rules. |
| enableSWRImage Access | Boolean | Whether a cluster is interconnected with SWR. This parameter is used and returned only by the Autopilot cluster creation API. To ensure that your cluster nodes can pull images from SWR, the existing SWR and OBS endpoints in the selected VPC are used by default. If not, new SWR and OBS endpoints will be automatically created. |
| enableAutopilot | Boolean | Whether the cluster is an Autopilot cluster |
| ipv6enable | Boolean | Whether a cluster uses the IPv6 mode. This parameter is not supported by Autopilot clusters. |
| hostNetwork | AutopilotHostNetwork object | Node networking parameters, including VPC and subnet ID. This field is mandatory because nodes in a cluster communicate with each other using a VPC. |
| containerNetwork | AutopilotContainerNetwork object | Container networking parameters, including the container network model and container CIDR block |
| eniNetwork | AutopilotEniNetwork object | Cloud Native 2.0 network settings |
| serviceNetwork | AutopilotServiceNetwork object | Service CIDR block, including IPv4 CIDR block |
| authentication | AutopilotAuthentication object | Configuration of the cluster authentication mode, which is not supported by Autopilot clusters |

| Parameter | Type | Description |
|------------------------|---|---|
| billingMode | Integer | Cluster billing mode <ul style="list-style-type: none"> • 0: pay-per-use This parameter defaults to pay-per-use . |
| kubernetesSvcIpRange | String | Service CIDR block or the IP address range which the kubernetes clusterIP must fall within. If this parameter is not specified during cluster creation, the default value 10.247.0.0/16 is used. This parameter is being deprecated, so you can use serviceNetwork instead. The new field contains the IPv4 CIDR blocks. |
| clusterTags | Array of AutopilotResourceTag objects | Cluster resource tags |
| kubeProxyMode | String | Service forwarding mode. <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. NOTE iptables is used by default. |
| az | String | AZ. This field is returned only for a query. For details about AZs supported by CCE, see Regions and Endpoints . |
| extendParam | AutopilotClusterExtendParam object | Extended field to decide whether a cluster will span across AZs or belong to a specified enterprise project, or whether a dedicated CCE cluster is to be created |
| configurationsOverride | Array of AutopilotPackageConfiguration objects | Overrides the default component configurations in a cluster. It is not supported in CCE Autopilot clusters. |

Table 4-98 AutopilotHostNetwork

| Parameter | Type | Description |
|-----------|--------|---|
| vpc | String | <p>ID of the VPC used to create a master node.</p> <p>Obtain the value using either of the following methods:</p> <ul style="list-style-type: none"> Method 1: Log in to the VPC console and view the VPC ID on the VPC details page. Method 2: Use the VPC API. For details, see Querying VPCs. |
| subnet | String | <p>Network ID of the subnet used to create a master node. Obtain the value using either of the following methods:</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. For details, see Querying Subnets. |

Table 4-99 AutopilotContainerNetwork

| Parameter | Type | Description |
|-----------|--------|---|
| mode | String | <p>Container network type</p> <ul style="list-style-type: none"> eni*: specifies the Cloud Native 2.0 network model. This model has integrated cloud native elastic network interfaces (ENIs), uses VPC CIDR blocks to allocate container IP addresses, and supports ELB passthrough networking for high network performance. Use this model when creating a cluster. |

Table 4-100 AutopilotEniNetwork

| Parameter | Type | Description |
|-----------|---|--|
| subnets | Array of AutopilotNetworkSubnet objects | IPv4 subnet ID list of the subnet where an ENI resides. Obtain the value using either of the following methods: <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 AutopilotNetworkSubnet

| Parameter | Type | Description |
|-----------|--------|--|
| subnetID | String | Specifies the IPv4 subnet ID of the subnet used to create control nodes and containers. (IPv6 is not supported.) To obtain the subnet ID, use either of the following methods: <ul style="list-style-type: none"> Method 1: Log in to the VPC console and click a subnet in the VPC. On the subnet details page, search for the IPv4 subnet ID. Method 2: Use the VPC API for obtaining subnets. For details, see Querying Subnets . |

Table 4-102 AutopilotServiceNetwork

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

Table 4-103 AutopilotAuthentication

| Parameter | Type | Description |
|-----------|--------|---|
| mode | String | Cluster authentication mode. The default value is rbac . |

Table 4-104 AutopilotResourceTag

| 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 4-105 AutopilotClusterExtendParam

| Parameter | Type | Description |
|---------------------|--------|--|
| enterpriseProjectId | String | ID of the enterprise project that a cluster belongs to NOTE <ul style="list-style-type: none"> An enterprise project can be configured only after the enterprise project function is enabled. |
| upgradeFrom | String | Records of how a cluster is upgraded to its current version |

Table 4-106 AutopilotPackageConfiguration

| Parameter | Type | Description |
|-----------|--------|----------------|
| name | String | Component name |

| Parameter | Type | Description |
|----------------|---|-------------------------------|
| configurations | Array of AutopilotConfigurationItem objects | Component configuration items |

Table 4-107 AutopilotConfigurationItem

| Parameter | Type | Description |
|-----------|--------|------------------------------------|
| name | String | Component configuration item name |
| value | Object | Component configuration item value |

Table 4-108 AutopilotClusterStatus

| 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 and needs to be manually deleted. ● 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. ● ResizeFailed: Cluster specification changing is abnormal. ● 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. ● Freezing: The cluster is being frozen. ● Frozen: The cluster has been frozen. ● UnFreezing: The cluster is being unfrozen. ● 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. You can manually delete the cluster. |

| Parameter | Type | Description |
|--------------|---|--|
| jobID | String | <p>ID of the task associated with the current cluster status. 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 being 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 | Why a cluster changes to the current state. This parameter is returned if the cluster is not in the Available state. |
| message | String | Detailed information about why a cluster changes to the current state. This parameter is returned if the cluster is not in the Available state. |
| endpoints | Array of AutopilotCluster-Endpoints objects | Access address of kube-apiserver in a cluster |
| isLocked | Boolean | CBC resource locked |
| lockScene | String | Scenario where the CBC resource is locked |
| lockSource | String | Resource locked |
| lockSourceId | String | ID of a locked resource |
| deleteOption | Object | Status of the configuration to be deleted. This parameter is contained only in the response to the deletion request. |
| deleteStatus | Object | Information of the status to be deleted. This parameter is contained only in the response to the deletion request. |

Table 4-109 AutopilotClusterEndpoints

| Parameter | Type | Description |
|-----------|--------|---|
| url | String | Access address of kube-apiserver in a 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 delivered.

```
{
  "kind": "Cluster",
  "apiVersion": "v3",
  "metadata": {
    "name": "test-cluster",
    "uid": "a736db34-2270-11ef-a160-0255ac100100",
    "creationTimestamp": "2024-06-04 12:47:50.886502 +0000 UTC",
    "updateTimestamp": "2024-06-04 13:24:08.809147153 +0000 UTC",
    "labels": {
      "FeatureGates": "arpOptimization,elbv3,xGPU"
    },
    "annotations": {
      "enableAutopilot": "true"
    },
    "alias": "test-cluster"
  },
  "spec": {
    "category": "Turbo",
    "type": "VirtualMachine",
    "flavor": "cce.autopilot.cluster",
    "version": "v1.28",
    "platformVersion": "cce.4.0",
    "hostNetwork": {
      "vpc": "c6549063-d459-4ae1-9550-b5fec6741b0f",
      "subnet": "3b18c2d5-b352-4f59-b421-c2d2d48a1333"
    },
    "containerNetwork": {
      "mode": "eni"
    },
    "eniNetwork": {
      "subnets": [ {
        "subnetID": "186f9322-50c5-4e5a-91e3-47da86959afc"
      } ]
    },
    "serviceNetwork": {
      "IPv4CIDR": "172.16.0.0/16"
    },
    "authentication": {
      "mode": "rbac"
    }
  }
}
```

```

    },
    "billingMode" : 0,
    "kubernetesSvcIpsRange" : "172.16.0.0/16",
    "kubeProxyMode" : "iptables",
    "az" : "cn-north-7c",
    "extendParam" : {
      "upgradefrom" : ""
    }
  },
  "status" : {
    "phase" : "Available",
    "jobID" : "ba0c981e-2275-11ef-b73b-0255ac100103",
    "endpoints" : [ {
      "url" : "https://a736db34-2270-11ef-a160-0255ac100100.cluster.cce.cn-
north-7.myhuaweicloud.com:5443",
      "type" : "Internal"
    } ],
    "deleteOption" : {
      "delete_eni" : "delete-block",
      "delete_net" : "delete-block"
    },
    "deleteStatus" : {
      "previous_total" : 47,
      "current_total" : 53,
      "updated" : 47,
      "added" : 6,
      "deleted" : 0
    }
  }
}
}
}

```

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 DeleteAutopilotClusterSolution {

    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();
DeleteAutopilotClusterRequest request = new DeleteAutopilotClusterRequest();
request.withClusterId("{cluster_id}");
try {
    DeleteAutopilotClusterResponse response = client.deleteAutopilotCluster(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

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["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 = DeleteAutopilotClusterRequest()
        request.cluster_id = "{cluster_id}"
        response = client.delete_autopilot_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.DeleteAutopilotClusterRequest{}
    request.ClusterId = "{cluster_id}"
    response, err := client.DeleteAutopilotCluster(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 delivered. |

Error Codes

See [Error Codes](#).

4.1.6 Obtaining a Cluster Certificate

Function

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

Constraints

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

Calling Method

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

URI

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

Table 4-110 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| project_id | Yes | String | <p>Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Project IDs of the account</p> <p>Default value: N/A</p> |
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |

Request Parameters

Table 4-111 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Table 4-112 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 4-113 Response header parameters

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

Table 4-114 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 |

| Parameter | Type | Description |
|-----------------|--------|---|
| current-context | String | Current context. If publicIp (VM EIP) exists, the value is external . If publicIp does not exist, the value is internal . |

Table 4-115 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 4-116 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 4-117 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 4-118 User

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

Table 4-119 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 4-120 Context

| Parameter | Type | Description |
|-----------|--------|-----------------|
| cluster | String | Cluster context |
| user | String | User context |

Example Requests

Apply for a cluster access certificate valid for 30 days.

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

Example Responses

Status code: 200

The certificate of the cluster is 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" : "Q2VydGlmaWNhdGU6*****FTkQgQ0VSVEIGSUNBVEUtLS0tLQo="
  }
}],
"users" : [ {
  "name" : "user",
  "user" : {
    "client-certificate-data" : "LS0tLS1CRUdJTiBDR*****QVRFLS0tLS0K",
    "client-key-data" : "LS0tLS1CRUdJTi*****BLRVktLS0tLQo="
  }
}],
"contexts" : [ {
  "name" : "internal",
  "context" : {
    "cluster" : "internalCluster",
    "user" : "user"
  }
}],
"current-context" : "internal"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Apply 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 CreateAutopilotKubernetesClusterCertSolution {

    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();
    }
}
```

```
        CreateAutopilotKubernetesClusterCertRequest request = new
CreateAutopilotKubernetesClusterCertRequest();
        request.withClusterId("{cluster_id}");
        CertDuration body = new CertDuration();
        body.withDuration(30);
        request.withBody(body);
        try {
            CreateAutopilotKubernetesClusterCertResponse response =
client.createAutopilotKubernetesClusterCert(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

Apply for a cluster access certificate valid for 30 days.

```
# coding: utf-8

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["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 = CreateAutopilotKubernetesClusterCertRequest()
        request.cluster_id = "{cluster_id}"
        request.body = CertDuration(
            duration=30
        )
        response = client.create_autopilot_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

Apply 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.CreateAutopilotKubernetesClusterCertRequest{}
    request.ClusterId = "{cluster_id}"
    request.Body = &model.CertDuration{
        Duration: int32(30),
    }
    response, err := client.CreateAutopilotKubernetesClusterCert(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 cluster is obtained. For details about the certificate file format, see the Kubernetes v1.Config structure. |

Error Codes

See [Error Codes](#).

4.1.7 Obtaining Job Information

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**, where **uri** specifies the resource 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 /autopilot/v3/projects/{project_id}/jobs/{job_id}

Table 4-121 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| project_id | Yes | String | <p>Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Project IDs of the account</p> <p>Default value: N/A</p> |
| job_id | Yes | String | <p>Job ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> |

Request Parameters

Table 4-122 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Response Parameters

Status code: 200

Table 4-123 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 4-124 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 |
| 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 4-125 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. |

| Parameter | Type | Description |
|-----------|------------------------------------|-------------------------|
| metadata | JobMetadata object | Job metadata |
| spec | JobSpec object | Detailed job parameters |
| status | JobStatus object | Job status |

Table 4-126 JobMetadata

| Parameter | Type | Description |
|-------------------|--------|-------------------------------|
| uid | String | Job ID |
| creationTimestamp | String | Time when the job was created |
| updateTimestamp | String | Time when the job was updated |

Table 4-127 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 job information is 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 ShowAutopilotJobSolution {

    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();
        ShowAutopilotJobRequest request = new ShowAutopilotJobRequest();
        request.withJobId("{job_id}");
        try {
            ShowAutopilotJobResponse response = client.showAutopilotJob(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

import os
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 = os.getenv("CLOUD_SDK_AK")
    sk = 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 = ShowAutopilotJobRequest()
    request.job_id = "{job_id}"
    response = client.show_autopilot_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.ShowAutopilotJobRequest{}
    request.JobId = "{job_id}"
    response, err := client.ShowAutopilotJob(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 information is obtained. |

Error Codes

See [Error Codes](#).

4.1.8 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**, where **uri** specifies the resource path for API access.

Calling Method

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

URI

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

Table 4-128 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| project_id | Yes | String | <p>Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Project IDs of the account</p> <p>Default value: N/A</p> |

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|--|
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |

Request Parameters

Table 4-129 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Table 4-130 Request body parameters

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

Table 4-131 MasterEIPRequestSpec

| Parameter | Mandatory | Type | Description |
|-----------|-----------|-----------------------------|--|
| action | No | String | <p>Binding or unbinding. Mandatory.</p> <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 4-132 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 4-133 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 API Server address of a cluster |
| status | status object | Status information |

Table 4-134 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 4-135 MasterEIPResponseSpec

| Parameter | Type | Description |
|-----------|--------|---------------------|
| action | String | A binding operation |

| Parameter | Type | Description |
|-----------|-----------------------------|--|
| spec | spec object | Configuration attributes of the elastic IP address to be bound |
| elasticIp | String | EIP |

Table 4-136 spec

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

Table 4-137 EipSpec

| Parameter | Type | Description |
|-----------|----------------------------------|-----------------------|
| bandwidth | bandwidth object | Bandwidth information |

Table 4-138 bandwidth

| Parameter | Type | Description |
|-----------|---------|----------------|
| size | Integer | Bandwidth size |
| sharetype | String | Bandwidth type |

Table 4-139 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-d7a22db0fd51"  
}  
}  
}
```

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 UpdateAutopilotClusterEipSolution {  
  
    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();
UpdateAutopilotClusterEipRequest request = new UpdateAutopilotClusterEipRequest();
request.withClusterId("{cluster_id}");
MasterEIPRequest body = new MasterEIPRequest();
MasterEIPRequestSpecSpec specSpec = new MasterEIPRequestSpecSpec();
specSpec.withId("a757a69e-f920-455a-b1ba-d7a22db0fd51");
MasterEIPRequestSpec specbody = new MasterEIPRequestSpec();
specbody.withAction(MasterEIPRequestSpec.ActionEnum.fromValue("bind"))
    .withSpec(specSpec);
body.withSpec(specbody);
request.withBody(body);
try {
    UpdateAutopilotClusterEipResponse response = client.updateAutopilotClusterEip(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

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["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 = UpdateAutopilotClusterEipRequest()
        request.cluster_id = "{cluster_id}"
        specSpec = MasterEIPRequestSpecSpec(
```

```
        id="a757a69e-f920-455a-b1ba-d7a22db0fd51"
    )
    specbody = MasterEIPRequestSpec(
        action="bind",
        spec=specSpec
    )
    request.body = MasterEIPRequest(
        spec=specbody
    )
    response = client.update_autopilot_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.UpdateAutopilotClusterEipRequest{}
    request.ClusterId = "{cluster_id}"
    idSpec := "a757a69e-f920-455a-b1ba-d7a22db0fd51"
    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.UpdateAutopilotClusterEip(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](#).

4.1.9 Obtaining Cluster Access Address

Function

This API is used to obtain the cluster access addresses by cluster ID, including the private IP and public IP.

NOTE

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

Calling Method

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

URI

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

Table 4-140 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| project_id | Yes | String | <p>Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Project IDs of the account</p> <p>Default value: N/A</p> |
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |

Request Parameters

Table 4-141 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Response Parameters

Status code: 200

Table 4-142 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 4-143 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 4-144 OpenAPISpec

| Parameter | Type | Description |
|-----------|--------------------|-----------------------------------|
| spec | spec object | Address for accessing the cluster |

Table 4-145 spec

| Parameter | Type | Description |
|-----------|-----------------------|-----------------------------|
| eip | EipSpec object | EIP details |
| IsDynamic | Boolean | Dynamic provisioning or not |

Table 4-146 EipSpec

| Parameter | Type | Description |
|-----------|-------------------------|-----------------------|
| bandwidth | bandwidth object | Bandwidth information |

Table 4-147 bandwidth

| Parameter | Type | Description |
|-----------|---------|----------------|
| size | Integer | Bandwidth size |
| sharetype | String | Bandwidth type |

Table 4-148 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.

```
{
  "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 ShowAutopilotClusterEndpointsSolution {

    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();
        ShowAutopilotClusterEndpointsRequest request = new ShowAutopilotClusterEndpointsRequest();
        request.withClusterId("{cluster_id}");
        try {
            ShowAutopilotClusterEndpointsResponse response = client.showAutopilotClusterEndpoints(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

import os
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 = os.environ["CLOUD_SDK_AK"]
sk = os.environ["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 = ShowAutopilotClusterEndpointsRequest()
    request.cluster_id = "{cluster_id}"
    response = client.show_autopilot_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.ShowAutopilotClusterEndpointsRequest{}
    request.ClusterId = "{cluster_id}"
    response, err := client.ShowAutopilotClusterEndpoints(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. |

Error Codes

See [Error Codes](#).

4.2 Add-on Management for Autopilot Clusters

4.2.1 Installing an Add-on Instance

Function

This API is used to install an add-on instance using an add-on template.

Calling Method

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

URI

POST /autopilot/v3/addons

Request Parameters

Table 4-149 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Table 4-150 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 4-151 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. |
| 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 4-152 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. |
| addonTemplateName | Yes | String | Name of the add-on template to be installed, for example, coredns . |

Response Parameters

Status code: 201

Table 4-153 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. |

| Parameter | Type | Description |
|-----------|--|--|
| spec | InstanceSpec object | Detailed description of the add-on instance. |
| status | AddonInstanceStatus object | Add-on instance status. |

Table 4-154 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 4-155 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. |

| Parameter | Type | Description |
|---------------------|---------------------|--|
| 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 4-156 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. |

| Parameter | Type | Description |
|-----------------|------------------------|--|
| 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. |
| previousVersion | String | The add-on version before upgrade or rollback |

Table 4-157 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 4-158 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. |

| Parameter | Type | Description |
|-----------|------------------|---|
| category | Array of strings | <p>The cluster types to which the settings are applied</p> <p>Options:</p> <ul style="list-style-type: none"> • CCE: CCE standard clusters • Turbo: CCE Turbo clusters • Autopilot: CCE Autopilot clusters <p>Default value:</p> <p>If this parameter is left blank, CCE standard and Turbo clusters are used by default.</p> |

Example Requests

Install CoreDNS 1.28.6 and set the number of add-on pods to 2.

```
{
  "kind": "Addon",
  "apiVersion": "v3",
  "metadata": {
    "annotations": {
      "addon.install/type": "install"
    }
  },
  "spec": {
    "clusterID": "597f2d95-44ab-11ef-9e39-0255ac100115",
    "version": "1.28.6",
    "addonTemplateName": "coredns",
    "values": {
      "basic": {
        "cluster_ip": "10.247.3.10",
        "image_version": "1.28.6",
        "swr_addr": "swr.cn-north-7.myhuaweicloud.com",
        "swr_user": "autopilot-official",
        "rbac_enabled": true,
        "cluster_version": "v1.28"
      },
      "flavor": {
        "category": [ "Autopilot" ],
        "is_default": true,
        "name": "autopilot-flavor1",
        "replicas": 2,
        "resources": [ {
          "id": "coredns",
          "name": "coredns",
          "limitsCpu": "1000m",
          "requestsCpu": "1000m",
          "limitsMem": "2048Mi",
          "requestsMem": "2048Mi"
        } ]
      },
      "custom": {
        "multiAZBalance": false,
        "multiAZEnabled": false,
        "node_match_expressions": [ ],
        "parameterSyncStrategy": "ensureConsistent",
        "servers": [ {
          "plugins": [ {
            "name": "bind",
```

```

    "parameters" : "${POD_IP}"
  }, {
    "configBlock" : "servfail 5s",
    "name" : "cache",
    "parameters" : 30
  }, {
    "name" : "errors"
  }, {
    "name" : "health",
    "parameters" : "${POD_IP}:8080"
  }, {
    "name" : "ready",
    "parameters" : "${POD_IP}:8081"
  }, {
    "configBlock" : "pods insecure\nfallthrough in-addr.arpa ip6.arpa",
    "name" : "kubernetes",
    "parameters" : "cluster.local in-addr.arpa ip6.arpa"
  }, {
    "name" : "loadbalance",
    "parameters" : "round_robin"
  }, {
    "name" : "prometheus",
    "parameters" : "${POD_IP}:9153"
  }, {
    "configBlock" : "policy random",
    "name" : "forward",
    "parameters" : ". /etc/resolv.conf"
  }, {
    "name" : "reload"
  } ],
  "port" : 5353,
  "zones" : [ {
    "zone" : "."
  } ]
}],
"stub_domains" : { },
"tolerations" : [ {
  "key" : "node.kubernetes.io/not-ready",
  "operator" : "Exists",
  "effect" : "NoExecute",
  "tolerationSeconds" : 60
}, {
  "key" : "node.kubernetes.io/unreachable",
  "operator" : "Exists",
  "effect" : "NoExecute",
  "tolerationSeconds" : 60
} ],
"upstream_nameservers" : [ ]
}
}
}
}

```

Example Responses

Status code: 201

OK

```

{
  "kind" : "Addon",
  "apiVersion" : "v3",
  "metadata" : {
    "uid" : "90b775e0-5774-4e1d-ab3b-516332ba047a",
    "name" : "coredns",
    "alias" : "coredns",
    "creationTimestamp" : "2024-07-18T04:04:21Z",
    "updateTimestamp" : "2024-07-18T04:04:21Z"
  },
}

```

```

"spec" : {
  "clusterID" : "597f2d95-44ab-11ef-9e39-0255ac100115",
  "version" : "1.28.6",
  "addonTemplateName" : "coredns",
  "addonTemplateType" : "helm",
  "addonTemplateLogo" : "",
  "addonTemplateLabels" : [ "ContainerNetwork" ],
  "description" : "CoreDNS is a DNS server that chains plugins and provides Kubernetes DNS Services",
  "values" : {
    "basic" : {
      "cluster_ip" : "10.247.3.10",
      "cluster_version" : "v1.28",
      "image_version" : "1.28.6",
      "platform" : "linux-amd64",
      "rbac_enabled" : true,
      "swr_addr" : "swr.cn-north-7.myhuaweicloud.com",
      "swr_user" : "autopilot-official"
    },
    "custom" : {
      "multiAZBalance" : false,
      "multiAZEnabled" : false,
      "node_match_expressions" : [ ],
      "parameterSyncStrategy" : "ensureConsistent",
      "servers" : [ {
        "plugins" : [ {
          "name" : "bind",
          "parameters" : "${POD_IP}"
        }, {
          "configBlock" : "servfail 5s",
          "name" : "cache",
          "parameters" : 30
        }, {
          "name" : "errors"
        }, {
          "name" : "health",
          "parameters" : "${POD_IP}:8080"
        }, {
          "name" : "ready",
          "parameters" : "${POD_IP}:8081"
        }, {
          "configBlock" : "pods insecure\nfallthrough in-addr.arpa ip6.arpa",
          "name" : "kubernetes",
          "parameters" : "cluster.local in-addr.arpa ip6.arpa"
        }, {
          "name" : "loadbalance",
          "parameters" : "round_robin"
        }, {
          "name" : "prometheus",
          "parameters" : "${POD_IP}:9153"
        }, {
          "configBlock" : "policy random",
          "name" : "forward",
          "parameters" : ". /etc/resolv.conf"
        }, {
          "name" : "reload"
        }
      ],
      "port" : 5353,
      "zones" : [ {
        "zone" : ""
      }
    ],
    "stub_domains" : { },
    "tolerations" : [ {
      "effect" : "NoExecute",
      "key" : "node.kubernetes.io/not-ready",
      "operator" : "Exists",
      "tolerationSeconds" : 60
    }, {
      "effect" : "NoExecute",

```

```

    "key" : "node.kubernetes.io/unreachable",
    "operator" : "Exists",
    "tolerationSeconds" : 60
  } ],
  "upstream_nameservers" : [ ]
},
"flavor" : {
  "category" : [ "Autopilot" ],
  "is_default" : true,
  "name" : "autopilot-flavor1",
  "replicas" : 2,
  "resources" : [ {
    "id" : "coredns",
    "limitsCpu" : "1000m",
    "limitsMem" : "2048Mi",
    "name" : "coredns",
    "requestsCpu" : "1000m",
    "requestsMem" : "2048Mi"
  } ]
},
"systemAutoInject" : {
  "cluster" : {
    "clusterID" : "597f2d95-44ab-11ef-9e39-0255ac100115",
    "clusterNetworkMode" : "eni",
    "clusterVersion" : "v1.28.5-r0"
  },
  "user" : {
    "projectID" : "47eb1d64cbeb45cfa01ae20af4f4b563"
  }
}
},
"status" : {
  "status" : "installing",
  "Reason" : "",
  "message" : "",
  "targetVersions" : null,
  "isRollbackable" : false,
  "currentVersion" : {
    "version" : "1.28.6",
    "input" : {
      "basic" : {
        "cluster_ip" : "10.247.3.10",
        "image_version" : "1.28.6",
        "swr_addr" : "swr.cn-north-7.myhuaweicloud.com",
        "swr_user" : "autopilot-official"
      }
    }
  },
  "parameters" : {
    "autopilot-flavor1" : {
      "category" : [ "Autopilot" ],
      "is_default" : true,
      "name" : "autopilot-flavor1",
      "replicas" : 2,
      "resources" : [ {
        "limitsCpu" : 1,
        "limitsMem" : "2Gi",
        "name" : "coredns",
        "requestsCpu" : 1,
        "requestsMem" : "2Gi"
      } ]
    }
  },
  "custom" : {
    "multiAZBalance" : false,
    "multiAZEnabled" : false,
    "node_match_expressions" : [ ],
    "parameterSyncStrategy" : "ensureConsistent",
    "servers" : [ {
      "plugins" : [ {
        "name" : "bind",

```



```

    "parameters" : "{$POD_IP}"
  }, {
    "configBlock" : "servfail 5s",
    "name" : "cache",
    "parameters" : 30
  }, {
    "name" : "errors"
  }, {
    "name" : "health",
    "parameters" : "{$POD_IP}:8080"
  }, {
    "name" : "ready",
    "parameters" : "{$POD_IP}:8081"
  }, {
    "configBlock" : "pods insecure\nfallthrough in-addr.arpa ip6.arpa",
    "name" : "kubernetes",
    "parameters" : "cluster.local in-addr.arpa ip6.arpa"
  }, {
    "name" : "loadbalance",
    "parameters" : "round_robin"
  }, {
    "name" : "prometheus",
    "parameters" : "{$POD_IP}:9153"
  }, {
    "configBlock" : "policy random",
    "name" : "forward",
    "parameters" : ". /etc/resolv.conf"
  }, {
    "name" : "reload"
  } ],
  "port" : 5353,
  "zones" : [ {
    "zone" : "."
  } ]
} ],
"stub_domains" : { },
"tolerations" : [ {
  "effect" : "NoExecute",
  "key" : "node.kubernetes.io/not-ready",
  "operator" : "Exists",
  "tolerationSeconds" : 60
}, {
  "effect" : "NoExecute",
  "key" : "node.kubernetes.io/unreachable",
  "operator" : "Exists",
  "tolerationSeconds" : 60
} ],
"upstream_nameservers" : [ ]
},
"flavor1" : {
  "is_default" : true,
  "name" : 2500,
  "recommend_cluster_flavor_types" : [ "small" ],
  "replicas" : 2,
  "resources" : [ {
    "limitsCpu" : "500m",
    "limitsMem" : "512Mi",
    "name" : "coredns",
    "requestsCpu" : "500m",
    "requestsMem" : "512Mi"
  } ]
},
"flavor2" : {
  "name" : 5000,
  "recommend_cluster_flavor_types" : [ "medium" ],
  "replicas" : 2,
  "resources" : [ {
    "limitsCpu" : "1000m",
    "limitsMem" : "1024Mi",

```

```

    "name" : "coredns",
    "requestsCpu" : "1000m",
    "requestsMem" : "1024Mi"
  } ]
},
"flavor3" : {
  "name" : 10000,
  "recommend_cluster_flavor_types" : [ "large" ],
  "replicas" : 2,
  "resources" : [ {
    "limitsCpu" : "2000m",
    "limitsMem" : "2048Mi",
    "name" : "coredns",
    "requestsCpu" : "2000m",
    "requestsMem" : "2048Mi"
  } ]
},
"flavor4" : {
  "name" : 20000,
  "recommend_cluster_flavor_types" : [ "xlarge" ],
  "replicas" : 4,
  "resources" : [ {
    "limitsCpu" : "2000m",
    "limitsMem" : "2048Mi",
    "name" : "coredns",
    "requestsCpu" : "2000m",
    "requestsMem" : "2048Mi"
  } ]
}
},
"stable" : true,
"translate" : {
  "en_US" : {
    "addon" : {
      "changeLog" : "Support autopilot cluster",
      "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 (for example, **\"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" : "les spécifications du plugin peuvent être associées aux spécifications du cluster. le fuseau horaire du plug-in est le même que celui du noeud",
      "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": "Autopilot clusters are supported.",
      "description": "CoreDNS is a DNS server that chains plug-ins and offers DNS resolution for
      Kubernetes clusters."
    },
    "description": {
      "Parameters.custom.stub_domains": "A domain name server for a custom domain name in key-
      value pair. The key is a suffix of DNS domain name, and the value is one or more DNS IP addresses, for
      example, **acme.local -- 1.2.3.4,6.7.8.9**.",
      "Parameters.custom.upstream_nameservers": "Resolves all domain names except intra-cluster
      service domain names and custom domain names. The value can be one or more DNS IP addresses, for
      example, **\"8.8.8.8\", \"8.8.4.4\"**.",
      "Parameters.flavor1.description": "Concurrent domain name resolution capability - external
      domain name: 2500 qps, internal domain name: 10000 qps",
      "Parameters.flavor1.name": 2500,
      "Parameters.flavor2.description": "Concurrent domain name resolution capability - external
      domain name: 5000 qps, internal domain name: 20000 qps",
      "Parameters.flavor2.name": 5000,
      "Parameters.flavor3.description": "Concurrent domain name resolution capability - external
      domain name: 10000 qps, internal domain name: 40000 qps",
      "Parameters.flavor3.name": 10000,
      "Parameters.flavor4.description": "Concurrent domain name resolution capability - 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 DNS server"
    }
  }
},
"supportVersions": null,
"creationTimestamp": "2024-02-19T11:33:46Z",
"updateTimestamp": "2024-02-21T01:24:05Z"
}
}
}

```

SDK Sample Code

The SDK sample code is as follows.

Java

Install CoreDNS 1.28.6 and set the number of add-on pods to 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 CreateAutopilotAddonInstanceSolution {

    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();

        CreateAutopilotAddonInstanceRequest request = new CreateAutopilotAddonInstanceRequest();
        InstanceRequest body = new InstanceRequest();
        Map<String, Object> listSpecValues = new HashMap<>();
        listSpecValues.put("basic", "{ \"cluster_version\": \"v1.28\", \"rbac_enabled\": true, \"swr_user\": \"autopilot-official\", \"image_version\": \"1.28.6\", \"cluster_ip\": \"10.247.3.10\", \"swr_addr\": \"swr.cn-north-7.myhuaweicloud.com\" }");
        listSpecValues.put("flavor", "{ \"replicas\": 2, \"name\": \"autopilot-flavor1\", \"resources\": [ { \"limitsCpu\": \"1000m\", \"name\": \"coredns\", \"id\": \"coredns\", \"limitsMem\": \"2048Mi\", \"requestsMem\": \"2048Mi\", \"requestsCpu\": \"1000m\" } ], \"category\": [ \"Autopilot\" ], \"is_default\": true }");
        listSpecValues.put("custom", "{ \"servers\": [ { \"port\": 5353, \"plugins\": [ { \"name\": \"bind\", \"parameters\": { \"POD_IP\" } }, { \"configBlock\": \"servfail 5s\", \"name\": \"cache\", \"parameters\": { \"errors\" }, { \"name\": \"health\", \"parameters\": { \"POD_IP\": 8080 } }, { \"name\": \"ready\", \"parameters\": { \"POD_IP\": 8081 } }, { \"configBlock\": \"pods insecure\\nfallthrough in-addr.arpa ip6.arpa\", \"name\": \"kubernetes\", \"parameters\": { \"cluster.local in-addr.arpa ip6.arpa\" }, { \"name\": \"loadbalance\", \"parameters\": { \"round_robin\" }, { \"name\": \"prometheus\", \"parameters\": { \"POD_IP\": 9153 } }, { \"configBlock\": \"policy random\", \"name\": \"forward\", \"parameters\": { \" /etc/resolv.conf\" }, { \"name\": \"reload\" } ], \"zones\": [ { \"zone\": \".\" } ] ], \"tolerations\": [ { \"effect\": \"NoExecute\", \"tolerationSeconds\": 60, \"key\": \"node.kubernetes.io/not-ready\", \"operator\": \"Exists\" }, { \"effect\": \"NoExecute\", \"tolerationSeconds\": 60, \"key\": \"node.kubernetes.io/unreachable\", \"operator\": \"Exists\" } ], \"multiAZBalance\": false, \"node_match_expressions\": [], \"stub_domains\": {}, \"multiAZEnabled\": false, \"parameterSyncStrategy\": \"ensureConsistent\", \"upstream_nameservers\": [] }");
        InstanceRequestSpec specbody = new InstanceRequestSpec();
        specbody.withVersion("1.28.6")
            .withClusterID("597f2d95-44ab-11ef-9e39-0255ac100115")
            .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 {
    CreateAutopilotAddonInstanceResponse response = client.createAutopilotAddonInstance(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 CoreDNS 1.28.6 and set the number of add-on pods to 2.

```

# coding: utf-8

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = CreateAutopilotAddonInstanceRequest()
        listValuesSpec = {
            "basic": "{ \"cluster_version\": \"v1.28\", \"rbac_enabled\": true, \"swr_user\": \"autopilot-official\", \"image_version\": \"1.28.6\", \"cluster_ip\": \"10.247.3.10\", \"swr_addr\": \"swr.cn-north-7.myhuaweicloud.com\" }",
            "flavor": "{ \"replicas\": 2, \"name\": \"autopilot-flavor1\", \"resources\": [{ \"limitsCpu\": \"1000m\", \"name\": \"coredns\", \"id\": \"coredns\", \"limitsMem\": \"2048Mi\", \"requestsMem\": \"2048Mi\", \"requestsCpu\": \"1000m\" }], \"category\": [\"Autopilot\"], \"is_default\": true }",
            "custom": "{ \"servers\": [{ \"port\": 5353, \"plugins\": [{ \"name\": \"bind\", \"parameters\": \"${POD_IP}\" }], \"configBlock\": \"servfail 5s\", \"name\": \"cache\", \"parameters\": 30, { \"name\": \"errors\", { \"name\": \"health\", \"parameters\": \"${POD_IP}:8080\", { \"name\": \"ready\", \"parameters\": \"${POD_IP}:8081\", { \"configBlock\": \"pods insecure\\nfallthrough in-addr.arpa ip6.arpa\", \"name\": \"kubernetes\", \"parameters\": \"cluster.local in-addr.arpa ip6.arpa\", \"name\": \"loadbalance\", \"parameters\": \"round_robin\", { \"name\": \"prometheus\", \"parameters\": \"${POD_IP}:9153\", { \"configBlock\": \"policy random\", \"name\": \"forward\", \"parameters\": \"/etc/resolv.conf\", { \"name\": \"reload\", \"zones\": [{ \"zone\": \".\" }], \"tolerations\": [{ \"effect\": \"NoExecute\", \"tolerationSeconds\": 60, \"key\": \"node.kubernetes.io/not-ready\", \"operator\": \"Exists\" }, { \"effect\": \"NoExecute\", \"tolerationSeconds\": 60, \"key\": \"node.kubernetes.io/unreachable\", \"operator\": \"Exists\" }], \"multiAZBalance\": false, \"node_match_expressions\": [], \"stub_domains\": { }, \"multiAZEnabled\": false, \"parameterSyncStrategy\": \"ensureConsistent\", \"upstream_nameservers\": [] } }"
        }
    }

```

```

specbody = InstanceRequestSpec(
    version="1.28.6",
    cluster_id="597f2d95-44ab-11ef-9e39-0255ac100115",
    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_autopilot_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 CoreDNS 1.28.6 and set the number of add-on pods to 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.CreateAutopilotAddonInstanceRequest{}
    var listValuesSpec = map[string]interface{}{
        "basic": "{ \"cluster_version\": \"v1.28\", \"rbac_enabled\": true, \"swr_user\": \"autopilot-official\", \"image_version\": \"1.28.6\", \"cluster_ip\": \"10.247.3.10\", \"swr_addr\": \"swr.cn-north-7.myhuaweicloud.com\" }",
        "flavor": "{ \"replicas\": 2, \"name\": \"autopilot-flavor1\", \"resources\": [ { \"limitsCpu\": \"1000m\", \"name\": \"coredns\", \"id\": \"coredns\", \"limitsMem\": \"2048Mi\", \"requestsMem\": \"2048Mi\", \"requestsCpu\": \"1000m\" } ], \"category\": [ \"Autopilot\" ], \"is_default\": true }",
        "custom": "{ \"servers\": [ { \"port\": 5353, \"plugins\": [ { \"name\": \"bind\", \"parameters\": \"${POD_IP}\" },

```

```

{"configBlock":{"servfail 5s","name":"cache","parameters":30},{"name":"errors"},{"name
":"health","parameters":{"POD_IP}:8080"},{"name":"ready","parameters":{"POD_IP}:8081"},
{"configBlock":{"pods insecure\nfallthrough in-addr.arpa ip6.arpa","name":"kubernetes","parameters
":"cluster.local in-addr.arpa ip6.arpa"},{"name":"loadbalance","parameters":{"round_robin"},{"name
":"prometheus","parameters":{"POD_IP}:9153"},{"configBlock":{"policy random","name":"forward
","parameters":{"/etc/resolv.conf"},{"name":"reload"}},{"zones":{"zone":"."}},"tolerations":
[{"effect":"NoExecute","tolerationSeconds":60,"key":"node.kubernetes.io/not-ready","operator
":"Exists"},{"effect":"NoExecute","tolerationSeconds":60,"key":"node.kubernetes.io/unreachable
","operator":"Exists"},{"multiAZBalance":false,"node_match_expressions":"","stub_domains":
{},"multiAZEnabled":false,"parameterSyncStrategy":"ensureConsistent","upstream_nameservers":[]},
}
}
versionSpec:= "1.28.6"
specbody := &model.InstanceRequestSpec{
    Version: &versionSpec,
    ClusterID: "597f2d95-44ab-11ef-9e39-0255ac100115",
    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.CreateAutopilotAddonInstance(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](#).

4.2.2 Listing Add-on Templates

Function

This API is used to obtain an add-on template.

Calling Method

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

URI

GET /autopilot/v3/addontemplates

Table 4-159 Query Parameters

| Parameter | Mandatory | Type | Description |
|---------------------|-----------|--------|---|
| addon_template_name | No | String | Add-on name or alias. If this parameter is not specified, all the add-ons will be obtained. |

Request Parameters

Table 4-160 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Response Parameters

Status code: 200

Table 4-161 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 4-162 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. |

| Parameter | Type | Description |
|-----------|-------------------------------------|--|
| spec | Templatespec object | Detailed description of the add-on template. |

Table 4-163 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 4-164 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 4-165 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 4-166 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. |
| category | Array of strings | The cluster types to which the settings are applied Options: <ul style="list-style-type: none"> • CCE: CCE standard clusters • Turbo: CCE Turbo clusters • Autopilot: CCE Autopilot clusters Default value: If this parameter is left blank, CCE standard and Turbo clusters are used by default. |

Example Requests

None

Example Responses

Status code: 200

OK

```

{
  "kind": "Addon",
  "apiVersion": "v3",
  "items": [ {
    "kind": "Addon",
    "apiVersion": "v3",
    "metadata": {
      "uid": "cie-collector",
      "name": "cie-collector",
      "alias": "kube-prometheus-stack",
      "creationTimestamp": "2024-01-26T09:06:25Z",
      "updateTimestamp": "2024-01-26T09:06:26Z"
    },
    "spec": {
      "type": "helm",
      "labels": [ "CloudNativeObservability" ],
      "description": "kube-prometheus-stack collects Kubernetes manifests, Prometheus rules combined with documentation and scripts to provide easy to operate end-to-end Kubernetes cluster monitoring with Prometheus using the Prometheus Operator.",
      "versions": [ {
        "version": "3.9.3",
        "input": {
          "basic": {
            "aom_url": "100.79.29.98:8149",
            "region_id": "cn-north-7",
            "swr_addr": "swr.cn-north-7.myhuaweicloud.com",
            "swr_user": "autopilot-official"
          },
          "parameters": {
            "autopilot-flavor1": {
              "category": [ "Autopilot" ],
              "deploy_mode": "server",
              "description": "Recommended when the number of containers in the cluster does not exceed
2000.",
              "name": "Autopilot-Small(<=2000 containers)",
              "resources": [ {
                "limitsCpu": "500m",
                "limitsMem": "1Gi",
                "name": "prometheusOperator"
              }, {
                "limitsCpu": "8",
                "limitsMem": "32Gi",
                "name": "prometheus"
              }, {
                "limitsCpu": "1",
                "limitsMem": "2Gi",
                "name": "thanosSidecar"
              }, {
                "limitsCpu": "4",
                "limitsMem": "16Gi",
                "name": "thanosQuery"
              }, {
                "limitsCpu": "4",
                "limitsMem": "16Gi",
                "name": "adapter"
              }, {
                "limitsCpu": "500m",
                "limitsMem": "1Gi",
                "name": "kubeStateMetrics"
              } ]
            },
            "autopilot-flavor2": {
              "category": [ "Autopilot" ],
              "deploy_mode": "server",
              "description": "Recommended when the number of containers in the cluster does not exceed
5000.",
              "name": "Autopilot-Medium(<=5000 containers)",
              "resources": [ {
                "limitsCpu": "500m",

```

```

        "limitsMem" : "1Gi",
        "name" : "prometheusOperator"
    }, {
        "limitsCpu" : "16",
        "limitsMem" : "64Gi",
        "name" : "prometheus"
    }, {
        "limitsCpu" : "2",
        "limitsMem" : "4Gi",
        "name" : "thanosSidecar"
    }, {
        "limitsCpu" : "8",
        "limitsMem" : "32Gi",
        "name" : "thanosQuery"
    }, {
        "limitsCpu" : "4",
        "limitsMem" : "32Gi",
        "name" : "adapter"
    }, {
        "limitsCpu" : "1",
        "limitsMem" : "2Gi",
        "name" : "kubeStateMetrics"
    }
    ]
},
"autopilot-flavor4" : {
    "category" : [ "Autopilot" ],
    "deploy_mode" : "server",
    "description" : "Custom configuration for this addon.",
    "name" : "custom-resources-autopilot-server",
    "resources" : [ {
        "limitsCpu" : "500m",
        "limitsMem" : "1Gi",
        "name" : "prometheusOperator"
    }, {
        "limitsCpu" : "16",
        "limitsMem" : "64Gi",
        "name" : "prometheus"
    }, {
        "limitsCpu" : "2",
        "limitsMem" : "4Gi",
        "name" : "thanosSidecar"
    }, {
        "limitsCpu" : "8",
        "limitsMem" : "32Gi",
        "name" : "thanosQuery"
    }, {
        "limitsCpu" : "4",
        "limitsMem" : "32Gi",
        "name" : "adapter"
    }, {
        "limitsCpu" : "1",
        "limitsMem" : "2Gi",
        "name" : "kubeStateMetrics"
    }
    ]
},
"autopilot-flavor5" : {
    "category" : [ "Autopilot" ],
    "deploy_mode" : "agent",
    "description" : "Recommended flavor for agent mode when the number of containers in the
cluster does not exceed 2000.",
    "is_default" : true,
    "name" : "Autopilot-Agent-Small(<=2000 containers)",
    "resources" : [ {
        "limitsCpu" : "1",
        "limitsMem" : "1Gi",
        "name" : "prometheusOperator",
        "requestsCpu" : "1",
        "requestsMem" : "1Gi"
    }, {

```

```

    "limitsCpu" : "1800m",
    "limitsMem" : "2900Mi",
    "name" : "prometheus",
    "requestsCpu" : "1800m",
    "requestsMem" : "2900Mi"
  }, {
    "limitsCpu" : "1",
    "limitsMem" : "1Gi",
    "name" : "kubeStateMetrics",
    "requestsCpu" : "1",
    "requestsMem" : "1Gi"
  }, {
    "limitsMem" : "500Mi",
    "name" : "nodeExporter",
    "requestsMem" : "100m"
  }
} ]
},
"autopilot-flavor6" : {
  "category" : [ "Autopilot" ],
  "deploy_mode" : "agent",
  "description" : "Recommended flavor for agent mode when the number of containers in the
cluster does not exceed 5000.",
  "name" : "Autopilot-Agent-Medium(<=5000 containers)",
  "resources" : [ {
    "limitsCpu" : "500m",
    "limitsMem" : "1Gi",
    "name" : "prometheusOperator"
  }, {
    "limitsCpu" : "4",
    "limitsMem" : "8Gi",
    "name" : "prometheus"
  }, {
    "limitsCpu" : "500m",
    "limitsMem" : "1Gi",
    "name" : "kubeStateMetrics"
  }
} ]
},
"autopilot-flavor8" : {
  "category" : [ "Autopilot" ],
  "deploy_mode" : "agent",
  "description" : "Custom flavor for agent mode",
  "name" : "custom-resources-autopilot-agent",
  "resources" : [ {
    "limitsCpu" : "500m",
    "limitsMem" : "1Gi",
    "name" : "prometheusOperator"
  }, {
    "limitsCpu" : "4",
    "limitsMem" : "8Gi",
    "name" : "prometheus"
  }, {
    "limitsCpu" : "500m",
    "limitsMem" : "1Gi",
    "name" : "kubeStateMetrics"
  }
} ]
},
"custom" : {
  "aom_app_key" : "",
  "aom_app_secret" : "",
  "aom_asm_app_key" : "",
  "aom_asm_app_secret" : "",
  "aom_asm_enable" : false,
  "aom_asm_insecure_skip_verify" : true,
  "aom_asm_instance_id" : "",
  "aom_asm_keep_regexp" : "istio.*",
  "aom_asm_project_id" : "",
  "aom_asm_remote_write_url" : "",
  "aom_auth_type" : "Bearer",
  "aom_enable" : false,

```

```

"aom_insecure_skip_verify" : true,
"aom_instance_id" : "",
"aom_project_id" : "",
"aom_region_id" : "",
"aom_remote_read_url" : "",
"aom_remote_write_url" : "",
"appCode" : "",
"appConfig" : {
  "adapter" : {
    "nodeAffinity_key" : "",
    "nodeAffinity_values" : "",
    "tolerations_effect" : "NoSchedule",
    "tolerations_key" : "",
    "tolerations_operator" : "Exists"
  },
  "alertmanager" : {
    "nodeAffinity_key" : "",
    "nodeAffinity_values" : "",
    "tolerations_effect" : "NoSchedule",
    "tolerations_key" : "",
    "tolerations_operator" : "Exists"
  },
  "kubeEventExporter" : {
    "nodeAffinity_key" : "",
    "nodeAffinity_values" : "",
    "tolerations_effect" : "NoSchedule",
    "tolerations_key" : "",
    "tolerations_operator" : "Exists"
  },
  "kubeStateMetrics" : {
    "nodeAffinity_key" : "",
    "nodeAffinity_values" : "",
    "tolerations_effect" : "NoSchedule",
    "tolerations_key" : "",
    "tolerations_operator" : "Exists"
  },
  "prometheusOperator" : {
    "nodeAffinity_key" : "",
    "nodeAffinity_values" : "",
    "tolerations_effect" : "NoSchedule",
    "tolerations_key" : "",
    "tolerations_operator" : "Exists"
  },
  "prometheusServer" : {
    "nodeAffinity_key" : "",
    "nodeAffinity_values" : "",
    "tolerations_effect" : "NoSchedule",
    "tolerations_key" : "",
    "tolerations_operator" : "Exists"
  },
  "thanosQuery" : {
    "nodeAffinity_key" : "",
    "nodeAffinity_values" : "",
    "tolerations_effect" : "NoSchedule",
    "tolerations_key" : "",
    "tolerations_operator" : "Exists"
  }
},
"basic_auth_password_third" : "",
"basic_auth_username_third" : "",
"bearer_token" : "",
"caCert" : "",
"certFile" : "",
"cieInstanceld" : "",
"cie_url" : "",
"cluster" : "",
"clusterId" : "",
"cluster_category" : "CCE",
"crd_install" : true,

```

```

"deploy_mode" : "agent",
"emptydir" : {
  "enabled" : false,
  "sizeLimit" : "10G"
},
"enableGcrypto" : true,
"enableRemote" : false,
"enableRemoteWrite" : false,
"enable_autopilot" : true,
"enable_cpd" : true,
"enable_custom_metrics" : true,
"enable_grafana" : true,
"enable_nodeAffinity" : false,
"enable_tolerations" : false,
"enablethird" : false,
"grafana_pvc_exist" : false,
"highAvailability" : false,
"insecureSkipVerify" : false,
"insecure_skip_verify_third" : false,
"install_with_cluster" : false,
"keyFile" : "",
"ksm_shards" : 1,
"lightweight" : true,
"lightweight_sts" : true,
"lightweight_sts_use_pvc" : false,
"local_hpa" : false,
"max_wal_time" : "30m",
"min_wal_time" : "1m",
"nodeAffinity_key" : "",
"nodeAffinity_values" : "",
"projectId" : "",
"region" : "cn-north-7",
"retention" : "1d",
"scrapeInterval" : "15s",
"scrape_insecure_skip_verify" : true,
"shards" : 1,
"storage_class" : "csi-disk-topology",
"storage_size" : "10Gi",
"storage_type" : "SAS",
"subnetId" : "",
"supportServerModeSharding" : false,
"tolerations_effect" : "NoSchedule",
"tolerations_key" : "",
"tolerations_operator" : "Exists",
"truncate_frequency" : "30m",
"url_third" : ""
}
},
"scenarios" : [ {
  "category" : [ "Autopilot" ],
  "custom" : {
    "cluster_category" : "Autopilot",
    "enableGcrypto" : false,
    "enable_cpd" : false,
    "storage_class" : "csi-disk"
  },
  "name" : "autopilot-config"
} ]
},
"stable" : true,
"translate" : {
  "en_US" : {
    "addon" : {
      "changeLog" : "The Autopilot cluster is supported.",
      "description" : "kube-prometheus-stack collects Kubernetes manifests, Grafana dashboards, and Prometheus rules combined with documentation and scripts to provide easy to operate end-to-end Kubernetes cluster monitoring with Prometheus using the Prometheus Operator. *Attention:kube-prometheus-stack is system monitoring component, When resources are insufficient, Kubernetes preferentially ensures pod scheduling."
    }
  }
}

```



```

    },
    "description" : {
      "Parameters.autopilot-flavor1.description" : "Recommended when the number of containers in the
cluster does not exceed 2000.",
      "Parameters.autopilot-flavor1.name" : "Small(<=2000 containers)",
      "Parameters.autopilot-flavor2.description" : "Recommended when the number of containers in the
cluster does not exceed 5000.",
      "Parameters.autopilot-flavor2.name" : "Medium(<=5000 containers)",
      "Parameters.autopilot-flavor3.description" : "Recommended when the number of containers in the
cluster exceeds 5000.",
      "Parameters.autopilot-flavor3.name" : "Large(>5000 containers)",
      "Parameters.autopilot-flavor4.description" : "Custom configuration for this addon.",
      "Parameters.autopilot-flavor4.name" : "Custom",
      "Parameters.autopilot-flavor5.description" : "Recommended when the number of containers in the
cluster does not exceed 2000.",
      "Parameters.autopilot-flavor5.name" : "Small(<=2000 containers)",
      "Parameters.autopilot-flavor6.description" : "Recommended when the number of containers in the
cluster does not exceed 5000.",
      "Parameters.autopilot-flavor6.name" : "Medium(<=5000 containers)",
      "Parameters.autopilot-flavor7.description" : "Recommended when the number of containers in the
cluster exceeds 5000.",
      "Parameters.autopilot-flavor7.name" : "Large(>5000 containers)",
      "Parameters.autopilot-flavor8.description" : "Custom configuration for this addon.",
      "Parameters.autopilot-flavor8.name" : "Custom",
      "Parameters.custom.deploy_mode" : "prometheus deploy mode",
      "Parameters.custom.highAvailability" : "high availability of prometheus and kube-event-exporter",
      "Parameters.custom.region" : "Availability region",
      "Parameters.custom.retention" : "Prometheus data retention period",
      "Parameters.custom.shards" : "Number of prometheus shards to distribute targets onto",
      "Parameters.custom.storage_size" : "Prometheus server data Persistent Volume size",
      "Parameters.custom.storage_type" : "Prometheus server data Persistent Volume Storage Class",
      "Parameters.custom.zone" : "Availability zone",
      "Parameters.flavor1.description" : "Just a demo for this addon. Recommended when the number
of containers in the cluster does not exceed 100.",
      "Parameters.flavor1.name" : "Demo(<=100 containers)",
      "Parameters.flavor2.description" : "Recommended when the number of containers in the cluster
does not exceed 2000.",
      "Parameters.flavor2.name" : "Small(<=2000 containers)",
      "Parameters.flavor3.description" : "Recommended when the number of containers in the cluster
does not exceed 5000.",
      "Parameters.flavor3.name" : "Medium(<=5000 containers)",
      "Parameters.flavor4.description" : "Recommended when the number of containers in the cluster
exceeds 5000.",
      "Parameters.flavor4.name" : "Large(>5000 containers)",
      "Parameters.flavor5.description" : "Custom configuration for this addon.",
      "Parameters.flavor5.name" : "custom-resources",
      "Parameters.flavor6.description" : "Default configuration for this addon.",
      "Parameters.flavor6.name" : "Default",
      "Parameters.flavor7.description" : "Custom configuration for this addon.",
      "Parameters.flavor7.name" : "Custom"
    },
    "key" : {
      "Parameters.custom.deploy_mode" : "prometheus deploy mode",
      "Parameters.custom.highAvailability" : "high availability",
      "Parameters.custom.region" : "availability region",
      "Parameters.custom.retention" : "data retention period",
      "Parameters.custom.storage_size" : "data Persistent Volume size",
      "Parameters.custom.storage_type" : "data Persistent Volume Storage Class",
      "Parameters.custom.zone" : "availability zone"
    }
  },
  "zh_CN" : {
    "addon" : {
      "changeLog" : "Autopilot clusters are supported.",
      "description" : "kube-prometheus-stack uses Prometheus-operator and Prometheus to provide
easy-to-use, end-to-end Kubernetes cluster monitoring capabilities. *Note: kube-prometheus-stack is a
system monitoring add-on. When cluster resources are insufficient, Kubernetes prioritizes the scheduling of
add-on pods."
    }
  },

```

```

        "description": {
            "Parameters.autopilot-flavor1.description": "Use this specification when the number of containers
in a cluster does not exceed 2,000.",
            "Parameters.autopilot-flavor1.name": "Small ( $\leq$  2,000 containers)",
            "Parameters.autopilot-flavor2.description": "Use this specification when the number of containers
in a cluster does not exceed 5,000.",
            "Parameters.autopilot-flavor2.name": "Medium ( $\leq$  5,000 containers)",
            "Parameters.autopilot-flavor3.description": "Use this specification when the number of containers
in a cluster exceeds 5,000.",
            "Parameters.autopilot-flavor3.name": "Large ( $>$  5,000 containers)",
            "Parameters.autopilot-flavor4.description": "Customize resource configurations.",
            "Parameters.autopilot-flavor4.name": "Custom",
            "Parameters.autopilot-flavor5.description": "Use this specification when the number of containers
in a cluster does not exceed 2,000.",
            "Parameters.autopilot-flavor5.name": "Small ( $\leq$  2,000 containers)",
            "Parameters.autopilot-flavor6.description": "Use this specification when the number of containers
in a cluster does not exceed 5,000.",
            "Parameters.autopilot-flavor6.name": "Medium ( $\leq$  5,000 containers)",
            "Parameters.autopilot-flavor7.description": "Use this specification when the number of containers
in a cluster exceeds 5,000.",
            "Parameters.autopilot-flavor7.name": "Large ( $>$  5,000 containers)",
            "Parameters.autopilot-flavor8.description": "Customize resource configurations.",
            "Parameters.autopilot-flavor8.name": "Custom",
            "Parameters.custom.deploy_mode": "Prometheus deployment mode",
            "Parameters.custom.highAvailability": "Deploy the Prometheus and Kubernetes event collection
and reporting services in HA mode. In this case, two available nodes are required to deploy instances with
two pods.",
            "Parameters.custom.region": "AZ",
            "Parameters.custom.retention": "Retention period of the Prometheus monitoring data",
            "Parameters.custom.shards": "Number of Prometheus service shards. Each shard is assigned a
different collection target.",
            "Parameters.custom.storage_size": "Persistent volume size of the Prometheus server",
            "Parameters.custom.storage_type": "Persistent volume types available for storing the Prometheus
data",
            "Parameters.custom.zone": "AZ",
            "Parameters.flavor1.description": "This specification is suitable for experience and function
demonstration environments where Prometheus occupies few resources and has limited processing
capabilities. Use this specification when the number of containers in a cluster does not exceed 100.",
            "Parameters.flavor1.name": "Demo ( $\leq$  100 containers)",
            "Parameters.flavor2.description": "Use this specification when the number of containers in a
cluster does not exceed 2,000.",
            "Parameters.flavor2.name": "Small ( $\leq$  2,000 containers)",
            "Parameters.flavor3.description": "Use this specification when the number of containers in a
cluster does not exceed 5,000.",
            "Parameters.flavor3.name": "Medium ( $\leq$  5,000 containers)",
            "Parameters.flavor4.description": "Use this specification when the number of containers in a
cluster exceeds 5,000.",
            "Parameters.flavor4.name": "Large ( $>$  5,000 containers)",
            "Parameters.flavor5.description": "Customize resource configurations.",
            "Parameters.flavor5.name": "Custom",
            "Parameters.flavor6.description": "Configure the default resource settings.",
            "Parameters.flavor6.name": "Default",
            "Parameters.flavor7.description": "Customize resource configurations.",
            "Parameters.flavor7.name": "Custom"
        },
        "key": {
            "Parameters.custom.deploy_mode": "Prometheus deployment mode",
            "Parameters.custom.highAvailability": "HA",
            "Parameters.custom.region": "AZ",
            "Parameters.custom.retention": "Data retention period",
            "Parameters.custom.storage_size": "Persistent volume size",
            "Parameters.custom.storage_type": "Persistent volume type",
            "Parameters.custom.zone": "AZ"
        }
    },
    "supportVersions": [ {
        "clusterType": "VirtualMachine",
        "clusterVersion": [ "v1.(27|28).*", "v1.(27|28).*" ],
    }
]

```

```
        "category" : [ "Autopilot" ]
    },
    "creationTimestamp" : "2024-01-26T09:06:25Z",
    "updateTimestamp" : "2024-01-26T09:06:25Z"
  }
}
}
```

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 ListAutopilotAddonTemplatesSolution {

    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();
        ListAutopilotAddonTemplatesRequest request = new ListAutopilotAddonTemplatesRequest();
        try {
            ListAutopilotAddonTemplatesResponse response = client.listAutopilotAddonTemplates(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
```

```
import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = ListAutopilotAddonTemplatesRequest()
        response = client.list_autopilot_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.ListAutopilotAddonTemplatesRequest{}
    response, err := client.ListAutopilotAddonTemplates(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](#).

4.2.3 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 /autopilot/v3/addons/{id}

Table 4-167 Path Parameters

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|--------------------|
| id | Yes | String | Add-on instance ID |

Request Parameters

Table 4-168 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Table 4-169 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 4-170 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. |
| 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 4-171 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. |
| addonTemplateName | Yes | String | Name of the add-on template to be installed, for example, coredns . |

Response Parameters

Status code: 200

Table 4-172 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. |

| Parameter | Type | Description |
|-----------|--|--|
| spec | InstanceSpec object | Detailed description of the add-on instance. |
| status | AddonInstanceStatus object | Add-on instance status. |

Table 4-173 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 4-174 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. |

| Parameter | Type | Description |
|---------------------|---------------------|--|
| 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 4-175 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. |

| Parameter | Type | Description |
|-----------------|------------------------|--|
| 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. |
| previousVersion | String | The add-on version before upgrade or rollback |

Table 4-176 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 4-177 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. |

| Parameter | Type | Description |
|-----------|------------------|---|
| category | Array of strings | <p>The cluster types to which the settings are applied</p> <p>Options:</p> <ul style="list-style-type: none"> • CCE: CCE standard clusters • Turbo: CCE Turbo clusters • Autopilot: CCE Autopilot clusters <p>Default value:</p> <p>If this parameter is left blank, CCE standard and Turbo clusters are used by default.</p> |

Example Requests

Update the CoreDNS add-on to version 1.28.6.

```
{
  "kind": "Addon",
  "apiVersion": "v3",
  "metadata": {
    "annotations": {
      "addon.upgrade/type": "upgrade"
    }
  },
  "spec": {
    "clusterID": "597f2d95-44ab-11ef-9e39-0255ac100115",
    "version": "1.28.6",
    "addonTemplateName": "coredns",
    "values": {
      "basic": {
        "cluster_ip": "10.247.3.10",
        "image_version": "1.28.6",
        "swr_addr": "swr.cn-north-7.myhuaweicloud.com",
        "swr_user": "autopilot-official",
        "rbac_enabled": true,
        "cluster_version": "v1.28"
      },
      "flavor": {
        "category": [ "Autopilot" ],
        "is_default": true,
        "name": "autopilot-flavor1",
        "replicas": 2,
        "resources": [ {
          "id": "coredns",
          "name": "coredns",
          "limitsCpu": "2000m",
          "requestsCpu": "2000m",
          "limitsMem": "2048Mi",
          "requestsMem": "2048Mi"
        } ]
      },
      "custom": {
        "multiAZBalance": false,
        "multiAZEnabled": false,
        "node_match_expressions": [ ],
        "parameterSyncStrategy": "ensureConsistent",
        "servers": [ {
          "plugins": [ {
            "name": "bind",
```



```

"updateTimestamp" : "2024-07-18T04:04:21Z"
},
"spec" : {
  "clusterID" : "597f2d95-44ab-11ef-9e39-0255ac100115",
  "version" : "1.28.6",
  "addonTemplateName" : "coredns",
  "addonTemplateType" : "helm",
  "addonTemplateLogo" : "",
  "addonTemplateLabels" : [ "ContainerNetwork" ],
  "description" : "CoreDNS is a DNS server that chains plugins and provides Kubernetes DNS Services",
  "values" : {
    "basic" : {
      "cluster_ip" : "10.247.3.10",
      "cluster_version" : "v1.28",
      "image_version" : "1.28.6",
      "rbac_enabled" : true,
      "swr_addr" : "swr.cn-north-7.myhuaweicloud.com",
      "swr_user" : "autopilot-official"
    },
    "custom" : {
      "extraConfig" : { },
      "multiAZBalance" : false,
      "multiAZEnabled" : false,
      "nodeSelector" : { },
      "node_match_expressions" : [ ],
      "parameterSyncStrategy" : "ensureConsistent",
      "servers" : [ {
        "plugins" : [ {
          "name" : "bind",
          "parameters" : "${POD_IP}"
        }, {
          "configBlock" : "servfail 5s",
          "name" : "cache",
          "parameters" : 30
        }, {
          "name" : "errors"
        }, {
          "name" : "health",
          "parameters" : "${POD_IP}:8080"
        }, {
          "name" : "ready",
          "parameters" : "${POD_IP}:8081"
        }, {
          "configBlock" : "pods insecure\nfallthrough in-addr.arpa ip6.arpa",
          "name" : "kubernetes",
          "parameters" : "cluster.local in-addr.arpa ip6.arpa"
        }, {
          "name" : "loadbalance",
          "parameters" : "round_robin"
        }, {
          "name" : "prometheus",
          "parameters" : "${POD_IP}:9153"
        }, {
          "configBlock" : "policy random",
          "name" : "forward",
          "parameters" : ". /etc/resolv.conf"
        }, {
          "name" : "reload"
        }
      ],
      "port" : 5353,
      "zones" : [ {
        "zone" : ""
      }
    ],
    "stub_domains" : { },
    "tolerations" : [ {
      "effect" : "NoExecute",
      "key" : "node.kubernetes.io/not-ready",
      "operator" : "Exists",

```

```

        "tolerationSeconds" : 60
    }, {
        "effect" : "NoExecute",
        "key" : "node.kubernetes.io/unreachable",
        "operator" : "Exists",
        "tolerationSeconds" : 60
    } ],
    "upstream_nameservers" : [ ]
},
"flavor" : {
    "category" : [ "Autopilot" ],
    "is_default" : true,
    "name" : "autopilot-flavor1",
    "replicas" : 2,
    "resources" : [ {
        "id" : "coredns",
        "limitsCpu" : "2000m",
        "limitsMem" : "2048Mi",
        "name" : "coredns",
        "requestsCpu" : "2000m",
        "requestsMem" : "2048Mi"
    } ]
},
"systemAutoInject" : {
    "cluster" : {
        "clusterID" : "597f2d95-44ab-11ef-9e39-0255ac100115",
        "clusterNetworkMode" : "enj",
        "clusterVersion" : "v1.28.5-r0"
    },
    "user" : {
        "projectID" : "47eb1d64cbeb45cfa01ae20af4f4b563"
    }
}
},
"status" : {
    "status" : "upgrading",
    "Reason" : "addon upgrading",
    "message" : "",
    "targetVersions" : null,
    "isRollbackable" : false,
    "currentVersion" : {
        "version" : "1.28.6",
        "input" : {
            "basic" : {
                "cluster_ip" : "10.247.3.10",
                "image_version" : "1.28.6",
                "swr_addr" : "swr.cn-north-7.myhuaweicloud.com",
                "swr_user" : "autopilot-official"
            }
        }
    },
    "parameters" : {
        "autopilot-flavor1" : {
            "category" : [ "Autopilot" ],
            "is_default" : true,
            "name" : "autopilot-flavor1",
            "replicas" : 2,
            "resources" : [ {
                "limitsCpu" : 1,
                "limitsMem" : "2Gi",
                "name" : "coredns",
                "requestsCpu" : 1,
                "requestsMem" : "2Gi"
            } ]
        }
    },
    "custom" : {
        "multiAZBalance" : false,
        "multiAZEnabled" : false,
        "node_match_expressions" : [ ],
        "parameterSyncStrategy" : "ensureConsistent",

```



```

"servers" : [ {
  "plugins" : [ {
    "name" : "bind",
    "parameters" : "{$POD_IP}"
  }, {
    "configBlock" : "servfail 5s",
    "name" : "cache",
    "parameters" : 30
  }, {
    "name" : "errors"
  }, {
    "name" : "health",
    "parameters" : "{$POD_IP}:8080"
  }, {
    "name" : "ready",
    "parameters" : "{$POD_IP}:8081"
  }, {
    "configBlock" : "pods insecure\nfallthrough in-addr.arpa ip6.arpa",
    "name" : "kubernetes",
    "parameters" : "cluster.local in-addr.arpa ip6.arpa"
  }, {
    "name" : "loadbalance",
    "parameters" : "round_robin"
  }, {
    "name" : "prometheus",
    "parameters" : "{$POD_IP}:9153"
  }, {
    "configBlock" : "policy random",
    "name" : "forward",
    "parameters" : ". /etc/resolv.conf"
  }, {
    "name" : "reload"
  } ],
  "port" : 5353,
  "zones" : [ {
    "zone" : ""
  } ]
} ],
"stub_domains" : { },
"tolerations" : [ {
  "effect" : "NoExecute",
  "key" : "node.kubernetes.io/not-ready",
  "operator" : "Exists",
  "tolerationSeconds" : 60
}, {
  "effect" : "NoExecute",
  "key" : "node.kubernetes.io/unreachable",
  "operator" : "Exists",
  "tolerationSeconds" : 60
} ],
"upstream_nameservers" : [ ]
},
"flavor1" : {
  "is_default" : true,
  "name" : 2500,
  "recommend_cluster_flavor_types" : [ "small" ],
  "replicas" : 2,
  "resources" : [ {
    "limitsCpu" : "500m",
    "limitsMem" : "512Mi",
    "name" : "coredns",
    "requestsCpu" : "500m",
    "requestsMem" : "512Mi"
  } ]
},
"flavor2" : {
  "name" : 5000,
  "recommend_cluster_flavor_types" : [ "medium" ],
  "replicas" : 2,

```

```

"resources" : [ {
  "limitsCpu" : "1000m",
  "limitsMem" : "1024Mi",
  "name" : "coredns",
  "requestsCpu" : "1000m",
  "requestsMem" : "1024Mi"
} ]
},
"flavor3" : {
  "name" : 10000,
  "recommend_cluster_flavor_types" : [ "large" ],
  "replicas" : 2,
  "resources" : [ {
    "limitsCpu" : "2000m",
    "limitsMem" : "2048Mi",
    "name" : "coredns",
    "requestsCpu" : "2000m",
    "requestsMem" : "2048Mi"
  } ]
},
"flavor4" : {
  "name" : 20000,
  "recommend_cluster_flavor_types" : [ "xlarge" ],
  "replicas" : 4,
  "resources" : [ {
    "limitsCpu" : "2000m",
    "limitsMem" : "2048Mi",
    "name" : "coredns",
    "requestsCpu" : "2000m",
    "requestsMem" : "2048Mi"
  } ]
}
}
},
"stable" : true,
"translate" : {
  "en_US" : {
    "addon" : {
      "changeLog" : "Support autopilot cluster",
      "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 (for example, **\"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 capability - external domain name: 2500 qps, internal domain name: 10000 qps",
      "Parameters.flavor1.name" : 2500,
      "Parameters.flavor2.description" : "Concurrent domain name resolution capability - external domain name: 5000 qps, internal domain name: 20000 qps",
      "Parameters.flavor2.name" : 5000,
      "Parameters.flavor3.description" : "Concurrent domain name resolution capability - external domain name: 10000 qps, internal domain name: 40000 qps",
      "Parameters.flavor3.name" : 10000,
      "Parameters.flavor4.description" : "Concurrent domain name resolution capability - 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" : "les spécifications du plugin peuvent être associées aux spécifications du cluster. le
fuseau horaire du plug-in est le même que celui du noeud",
    "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" : "Autopilot clusters are supported.",
    "description" : "CoreDNS is a DNS server that chains plug-ins and offers DNS resolution for
Kubernetes clusters."
  },
  "description" : {
    "Parameters.custom.stub_domains" : "A domain name server for a custom domain name in key-
value pair. The key is a suffix of DNS domain name, and the value is one or more DNS IP addresses, for
example, **acme.local -- 1.2.3.4,6.7.8.9**.",
    "Parameters.custom.upstream_nameservers" : "Resolves all domain names except intra-cluster
service domain names and custom domain names. The value can be one or more DNS IP addresses, for
example, **\"8.8.8.8\", \"8.8.4.4\"**.",
    "Parameters.flavor1.description" : "Concurrent domain name resolution capability - external
domain name: 2500 qps, internal domain name: 10,000 qps",
    "Parameters.flavor1.name" : 2500,
    "Parameters.flavor2.description" : "Concurrent domain name resolution capability - external
domain name: 5000 qps, internal domain name: 20000 qps",
    "Parameters.flavor2.name" : 5000,
    "Parameters.flavor3.description" : "Concurrent domain name resolution capability - external
domain name: 10000 qps, internal domain name: 40000 qps",
    "Parameters.flavor3.name" : 10000,
    "Parameters.flavor4.description" : "Concurrent domain name resolution capability - 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 DNS server"
  }
},
"supportVersions" : null,
"creationTimestamp" : "2024-02-19T11:33:46Z",
"updateTimestamp" : "2024-02-21T01:24:05Z"
}
}
}

```

SDK Sample Code

The SDK sample code is as follows.

Java

Update the CoreDNS add-on to version 1.28.6.

```
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 UpdateAutopilotAddonInstanceSolution {

    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();

        UpdateAutopilotAddonInstanceRequest request = new UpdateAutopilotAddonInstanceRequest();
        request.withId("{id}");
        InstanceRequest body = new InstanceRequest();
        Map<String, Object> listSpecValues = new HashMap<>();
        listSpecValues.put("basic", "{ \"cluster_version\": \"v1.28\", \"rbac_enabled\": true, \"swr_user\": \"autopilot-official\", \"image_version\": \"1.28.6\", \"cluster_ip\": \"10.247.3.10\", \"swr_addr\": \"swr.cn-north-7.myhuaweicloud.com\" }");
        listSpecValues.put("flavor", "{ \"replicas\": 2, \"name\": \"autopilot-flavor1\", \"resources\": { \"limitsCpu\": \"2000m\", \"name\": \"coredns\", \"id\": \"coredns\", \"limitsMem\": \"2048Mi\", \"requestsMem\": \"2048Mi\", \"requestsCpu\": \"2000m\" }, \"category\": [ \"Autopilot\" ], \"is_default\": true }");
        listSpecValues.put("custom", "{ \"extraConfig\": { }, \"servers\": { \"port\": 5353, \"plugins\": { \"name\": \"bind\", \"parameters\": { \"POD_IP\" }, \"configBlock\": \"servfail 5s\", \"name\": \"cache\", \"parameters\": { \"name\": \"errors\" }, \"name\": \"health\", \"parameters\": { \"POD_IP\": 8080 }, \"name\": \"ready\", \"parameters\": { \"POD_IP\": 8081 }, \"configBlock\": \"pods insecure\\nfallthrough in-addr.arpa ip6.arpa\", \"name\": \"kubernetes\", \"parameters\": \"cluster.local in-addr.arpa ip6.arpa\", \"name\": \"loadbalance\", \"parameters\": \"round_robin\", \"name\": \"prometheus\", \"parameters\": { \"POD_IP\": 9153 }, \"configBlock\": \"policy random\", \"name\": \"forward\", \"parameters\": \"/etc/resolv.conf\", \"name\": \"reload\" }, \"zones\": { \"zone\": \"\" }, \"tolerations\": { \"effect\": \"NoExecute\", \"tolerationSeconds\": 60, \"key\": \"node.kubernetes.io/not-ready\", \"operator\": \"Exists\", \"effect\": \"NoExecute\", \"tolerationSeconds\": 60, \"key\": \"node.kubernetes.io/unreachable\", \"operator\": \"Exists\" }, \"multiAZBalance\": false, \"node_match_expressions\": [ ], \"stub_domains\": { }, \"multiAZEnabled\": false, \"parameterSyncStrategy\": \"ensureConsistent\", \"upstream_nameservers\": [ ], \"nodeSelector\": { } }");
        InstanceRequestSpec specbody = new InstanceRequestSpec();
        specbody.withVersion("1.28.6")
            .withClusterID("597f2d95-44ab-11ef-9e39-0255ac100115")
            .withValues(listSpecValues)
            .withAddonTemplateName("coredns");
    }
}
```

```

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 {
    UpdateAutopilotAddonInstanceResponse response = client.updateAutopilotAddonInstance(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 CoreDNS add-on to version 1.28.6.

```

# coding: utf-8

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = UpdateAutopilotAddonInstanceRequest()
        request.id = "{id}"
        listValuesSpec = {
            "basic": {"cluster_version": "v1.28", "rbac_enabled": true, "swr_user": "autopilot-official",
            "image_version": "1.28.6", "cluster_ip": "10.247.3.10", "swr_addr": "swr.cn-north-7.myhuaweicloud.com"},
            "flavor": {"replicas": 2, "name": "autopilot-flavor1", "resources": {"limitsCpu": "2000m",
            "name": "coredns", "id": "coredns", "limitsMem": "2048Mi", "requestsMem": "2048Mi",
            "requestsCpu": "2000m"}, "category": ["Autopilot"], "is_default": true},
            "custom": {"extraConfig": {}, "servers": [{"port": 5353, "plugins": [{"name": "bind",
            "parameters": {"$POD_IP}], "configBlock": "servfail 5s", "name": "cache", "parameters": 30,
            "name": "errors", "name": "health", "parameters": {"$POD_IP": 8080}, "name": "ready",
            "parameters": {"$POD_IP": 8081}], "configBlock": "pods insecure\nfallthrough in-addr.arpa ip6.arpa",
            "name": "kubernetes", "parameters": "cluster.local in-addr.arpa ip6.arpa"}, {"name": "loadbalance

```

```

\,"parameters\":"round_robin"},{"name\":"prometheus","\parameters\":"${POD_IP}:9153"},
{"configBlock\":"policy_random","\name\":"forward","\parameters\":" /etc/resolv.conf"},{"name
\":"reload"},{"zones\":[{"zone\":"."}]}],{"tolerations\":[{"effect\":"NoExecute","\tolerationSeconds
\":"60","\key\":"node.kubernetes.io/not-ready","\operator\":"Exists"},{"effect\":"NoExecute
\":"tolerationSeconds\":"60","\key\":"node.kubernetes.io/unreachable","\operator\":"Exists
"}]}],{"multiAZBalance\":"false","\node_match_expressions\":"[]","\stub_domains\":"{}","\multiAZEnabled
\":"false","\parameterSyncStrategy\":"ensureConsistent","\upstream_nameservers\":"[]","\nodeSelector\":"{}"
}
}
specbody = InstanceRequestSpec(
    version="1.28.6",
    cluster_id="597f2d95-44ab-11ef-9e39-0255ac100115",
    values=listValuesSpec,
    addon_template_name="coredns"
)
listAnnotationsMetadata = {
    "addon.upgrade/type": "upgrade"
}
metadatabody = AddonMetadata(
    annotations=listAnnotationsMetadata
)
request.body = InstanceRequest(
    spec=specbody,
    metadata=metadatabody,
    api_version="v3",
    kind="Addon"
)
response = client.update_autopilot_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 CoreDNS add-on to version 1.28.6.

```

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.UpdateAutopilotAddonInstanceRequest{}

```

```

request.Id = "{id}"
var listValuesSpec = map[string]interface{}{
    "basic": "{ \"cluster_version\": \"v1.28\", \"rbac_enabled\": true, \"swr_user\": \"autopilot-official\", \"image_version\": \"1.28.6\", \"cluster_ip\": \"10.247.3.10\", \"swr_addr\": \"swr.cn-north-7.myhuaweicloud.com\" }",
    "flavor": "{ \"replicas\": 2, \"name\": \"autopilot-flavor1\", \"resources\": { \"limitsCpu\": \"2000m\", \"name\": \"coredns\", \"id\": \"coredns\", \"limitsMem\": \"2048Mi\", \"requestsMem\": \"2048Mi\", \"requestsCpu\": \"2000m\" } }, \"category\": [ \"Autopilot\" ], \"is_default\": true }",
    "custom": "{ \"extraConfig\": {}, \"servers\": [ { \"port\": 5353, \"plugins\": { \"name\": \"bind\", \"parameters\": { \"$POD_IP\" } }, \"configBlock\": \"servfail 5s\", \"name\": \"cache\", \"parameters\": { \"name\": \"errors\" } }, { \"name\": \"health\", \"parameters\": { \"$POD_IP\": 8080 } }, { \"name\": \"ready\", \"parameters\": { \"$POD_IP\": 8081 } }, { \"configBlock\": \"pods insecure\\nfallthrough in-addr.arpa ip6.arpa\", \"name\": \"kubernetes\", \"parameters\": \"cluster.local in-addr.arpa ip6.arpa\", \"name\": \"loadbalance\", \"parameters\": \"round-robin\", \"name\": \"prometheus\", \"parameters\": { \"$POD_IP\": 9153 } }, { \"configBlock\": \"policy random\", \"name\": \"forward\", \"parameters\": \"/etc/resolv.conf\", \"name\": \"reload\" } ], \"zones\": [ { \"zone\": \".\" } ] }, \"tolerations\": [ { \"effect\": \"NoExecute\", \"tolerationSeconds\": 60, \"key\": \"node.kubernetes.io/not-ready\", \"operator\": \"Exists\" }, { \"effect\": \"NoExecute\", \"tolerationSeconds\": 60, \"key\": \"node.kubernetes.io/unreachable\", \"operator\": \"Exists\" } ], \"multiAZBalance\": false, \"node_match_expressions\": [], \"stub_domains\": {}, \"multiAZEnabled\": false, \"parameterSyncStrategy\": \"ensureConsistent\", \"upstream_nameservers\": [], \"nodeSelector\": {} }",
}
versionSpec := "1.28.6"
specbody := &model.InstanceRequestSpec{
    Version: &versionSpec,
    ClusterID: "597f2d95-44ab-11ef-9e39-0255ac100115",
    Values: listValuesSpec,
    AddonTemplateName: "coredns",
}
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.UpdateAutopilotAddonInstance(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](#).

4.2.4 Rolling Back an Add-on Instance

Function

This API is used to roll back the version of an add-on instance to the source version. Only when **status.isRollbackable** 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 /autopilot/v3/addons/{id}/operation/rollback

Table 4-178 Path Parameters

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|--------------------|
| id | Yes | String | Add-on instance ID |

Request Parameters

Table 4-179 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Table 4-180 Request body parameters

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|-------------|
| clusterID | Yes | String | Cluster ID |

Response Parameters

Status code: 200

Table 4-181 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 4-182 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 4-183 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 4-184 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. |

| Parameter | Type | Description |
|-----------------|------------------------|--|
| 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. |
| previousVersion | String | The add-on version before upgrade or rollback |

Table 4-185 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 4-186 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. |

| Parameter | Type | Description |
|-----------|------------------|---|
| category | Array of strings | <p>The cluster types to which the settings are applied</p> <p>Options:</p> <ul style="list-style-type: none"> • CCE: CCE standard clusters • Turbo: CCE Turbo clusters • Autopilot: CCE Autopilot clusters <p>Default value:</p> <p>If this parameter is left blank, CCE standard and Turbo clusters are used by default.</p> |

Example Requests

```
{
  "clusterID" : "*****"
}
```

Example Responses

Status code: 200

The add-on instance version is rolled back.

```
{
  "kind" : "Addon",
  "apiVersion" : "v3",
  "metadata" : {
    "uid" : "4eba2678-330f-430b-aefc-821e1bbeff34",
    "name" : "coredns",
    "alias" : "coredns",
    "creationTimestamp" : "2024-07-18T03:03:17Z",
    "updateTimestamp" : "2024-07-18T03:04:44Z"
  },
  "spec" : {
    "clusterID" : "597f2d95-44ab-11ef-9e39-0255ac100115",
    "version" : "1.28.4",
    "addonTemplateName" : "coredns",
    "addonTemplateType" : "helm",
    "addonTemplateLogo" : "",
    "addonTemplateLabels" : [ "ContainerNetwork" ],
    "description" : "CoreDNS is a DNS server that chains plugins and provides Kubernetes DNS Services",
    "values" : {
      "basic" : {
        "cluster_ip" : "10.247.3.10",
        "cluster_version" : "v1.28",
        "image_version" : "1.28.4",
        "platform" : "linux-amd64",
        "rbac_enabled" : true,
        "swr_addr" : "swr.cn-north-7.myhuaweicloud.com",
        "swr_user" : "autopilot-official"
      },
      "custom" : {
        "extraConfig" : { },
        "multiAZBalance" : false,
        "multiAZEnabled" : false,
        "nodeSelector" : { },
        "node_match_expressions" : [ ],

```

```

"parameterSyncStrategy" : "ensureConsistent",
"servers" : [ {
  "plugins" : [ {
    "name" : "bind",
    "parameters" : "${POD_IP}"
  }, {
    "configBlock" : "servfail 5s",
    "name" : "cache",
    "parameters" : 30
  }, {
    "name" : "errors"
  }, {
    "name" : "health",
    "parameters" : "${POD_IP}:8080"
  }, {
    "name" : "ready",
    "parameters" : "${POD_IP}:8081"
  }, {
    "configBlock" : "pods insecure\nfallthrough in-addr.arpa ip6.arpa",
    "name" : "kubernetes",
    "parameters" : "cluster.local in-addr.arpa ip6.arpa"
  }, {
    "name" : "loadbalance",
    "parameters" : "round_robin"
  }, {
    "name" : "prometheus",
    "parameters" : "${POD_IP}:9153"
  }, {
    "configBlock" : "policy random",
    "name" : "forward",
    "parameters" : ". /etc/resolv.conf"
  }, {
    "name" : "reload"
  } ],
  "port" : 5353,
  "zones" : [ {
    "zone" : ""
  } ]
}],
"stub_domains" : { },
"tolerations" : [ {
  "effect" : "NoExecute",
  "key" : "node.kubernetes.io/not-ready",
  "operator" : "Exists",
  "tolerationSeconds" : 60
}, {
  "effect" : "NoExecute",
  "key" : "node.kubernetes.io/unreachable",
  "operator" : "Exists",
  "tolerationSeconds" : 60
}],
"upstream_nameservers" : [ ]
},
"flavor" : {
  "category" : [ "Autopilot" ],
  "is_default" : true,
  "name" : "autopilot-flavor1",
  "replicas" : 2,
  "resources" : [ {
    "id" : "coredns",
    "limitsCpu" : "1000m",
    "limitsMem" : "1024Mi",
    "name" : "coredns",
    "requestsCpu" : "1000m",
    "requestsMem" : "1024Mi"
  } ]
},
"image" : {
  "pullPolicy" : "Always"
}

```

```

},
"isClusterService" : true,
"multiAZPreferred" : {
  "podAntiAffinity" : {
    "preferredDuringSchedulingIgnoredDuringExecution" : [ {
      "podAffinityTerm" : {
        "labelSelector" : {
          "matchExpressions" : [ {
            "key" : "app",
            "operator" : "In",
            "values" : [ "coredns" ]
          } ]
        }
      }
    ]
  },
  "topologyKey" : "topology.kubernetes.io/zone"
},
"weight" : 100
} ]
},
"multiAZRequired" : {
  "podAntiAffinity" : {
    "requiredDuringSchedulingIgnoredDuringExecution" : [ {
      "labelSelector" : {
        "matchExpressions" : [ {
          "key" : "az-antiaffinity-app",
          "operator" : "In",
          "values" : [ "coredns" ]
        } ]
      }
    ]
  },
  "topologyKey" : "topology.kubernetes.io/zone"
} ]
},
"nodeSelector" : { },
"rbac" : {
  "create" : true,
  "serviceAccountName" : "default"
},
"service" : {
  "annotations" : {
    "prometheus.io/port" : "9153",
    "prometheus.io/scrape" : "true"
  },
  "clusterIP" : "10.247.3.10",
  "type" : "ClusterIP"
},
"systemAutoInject" : {
  "cluster" : {
    "clusterID" : "597f2d95-44ab-11ef-9e39-0255ac100115",
    "clusterNetworkMode" : "eni",
    "clusterVersion" : "v1.28.5-r0"
  },
  "user" : {
    "projectID" : "47eb1d64cbeb45cfa01ae20af4f4b563"
  }
},
"topologySpreadConstraints" : [ {
  "labelSelector" : {
    "matchLabels" : {
      "app" : "coredns"
    }
  },
  "maxSkew" : 1,
  "topologyKey" : "topology.kubernetes.io/zone",
  "whenUnsatisfiable" : "DoNotSchedule"
} ],
"zoneFiles" : [ ]
}
},

```

```

"status" : {
  "status" : "rollbacking",
  "Reason" : "Rollback to 1",
  "message" : "",
  "targetVersions" : [ "1.28.6" ],
  "isRollbackable" : false,
  "previousVersion" : "1.28.6",
  "currentVersion" : {
    "version" : "1.28.4",
    "input" : {
      "basic" : {
        "cluster_ip" : "10.247.3.10",
        "image_version" : "1.28.4",
        "swr_addr" : "swr.cn-north-7.myhuaweicloud.com",
        "swr_user" : "autopilot-official"
      },
      "parameters" : {
        "autopilot-flavor1" : {
          "category" : [ "Autopilot" ],
          "is_default" : true,
          "name" : "autopilot-flavor1",
          "replicas" : 2,
          "resources" : [ {
            "limitsCpu" : 1,
            "limitsMem" : "1Gi",
            "name" : "coredns",
            "requestsCpu" : 1,
            "requestsMem" : "1Gi"
          } ]
        }
      },
      "custom" : {
        "multiAZBalance" : false,
        "multiAZEnabled" : false,
        "node_match_expressions" : [ ],
        "parameterSyncStrategy" : "ensureConsistent",
        "servers" : [ {
          "plugins" : [ {
            "name" : "bind",
            "parameters" : "{$POD_IP}"
          }, {
            "configBlock" : "servfail 5s",
            "name" : "cache",
            "parameters" : 30
          }, {
            "name" : "errors"
          }, {
            "name" : "health",
            "parameters" : "{$POD_IP}:8080"
          }, {
            "name" : "ready",
            "parameters" : "{$POD_IP}:8081"
          }, {
            "configBlock" : "pods insecure\nfallthrough in-addr.arpa ip6.arpa",
            "name" : "kubernetes",
            "parameters" : "cluster.local in-addr.arpa ip6.arpa"
          }, {
            "name" : "loadbalance",
            "parameters" : "round_robin"
          }, {
            "name" : "prometheus",
            "parameters" : "{$POD_IP}:9153"
          }, {
            "configBlock" : "policy random",
            "name" : "forward",
            "parameters" : ". /etc/resolv.conf"
          }, {
            "name" : "reload"
          }
        ] ],
        "port" : 5353,

```



```

    "zones" : [ {
      "zone" : ""
    } ],
  },
  "stub_domains" : { },
  "tolerations" : [ {
    "effect" : "NoExecute",
    "key" : "node.kubernetes.io/not-ready",
    "operator" : "Exists",
    "tolerationSeconds" : 60
  }, {
    "effect" : "NoExecute",
    "key" : "node.kubernetes.io/unreachable",
    "operator" : "Exists",
    "tolerationSeconds" : 60
  } ],
  "upstream_nameservers" : [ ]
},
"flavor1" : {
  "is_default" : true,
  "name" : 2500,
  "recommend_cluster_flavor_types" : [ "small" ],
  "replicas" : 2,
  "resources" : [ {
    "limitsCpu" : "500m",
    "limitsMem" : "512Mi",
    "name" : "coredns",
    "requestsCpu" : "500m",
    "requestsMem" : "512Mi"
  } ]
},
"flavor2" : {
  "name" : 5000,
  "recommend_cluster_flavor_types" : [ "medium" ],
  "replicas" : 2,
  "resources" : [ {
    "limitsCpu" : "1000m",
    "limitsMem" : "1024Mi",
    "name" : "coredns",
    "requestsCpu" : "1000m",
    "requestsMem" : "1024Mi"
  } ]
},
"flavor3" : {
  "name" : 10000,
  "recommend_cluster_flavor_types" : [ "large" ],
  "replicas" : 2,
  "resources" : [ {
    "limitsCpu" : "2000m",
    "limitsMem" : "2048Mi",
    "name" : "coredns",
    "requestsCpu" : "2000m",
    "requestsMem" : "2048Mi"
  } ]
},
"flavor4" : {
  "name" : 20000,
  "recommend_cluster_flavor_types" : [ "xlarge" ],
  "replicas" : 4,
  "resources" : [ {
    "limitsCpu" : "2000m",
    "limitsMem" : "2048Mi",
    "name" : "coredns",
    "requestsCpu" : "2000m",
    "requestsMem" : "2048Mi"
  } ]
}
}
},

```

```

"stable" : true,
"translate" : {
  "en_US" : {
    "addon" : {
      "changeLog" : "plugin specifications can be associated with cluster specifications. The time zone of
the plug-in is the same as that of the node",
      "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 (for example, **\"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 capability - external
domain name: 2500 qps, internal domain name: 10000 qps",
      "Parameters.flavor1.name" : 2500,
      "Parameters.flavor2.description" : "Concurrent domain name resolution capability - external
domain name: 5000 qps, internal domain name: 20000 qps",
      "Parameters.flavor2.name" : 5000,
      "Parameters.flavor3.description" : "Concurrent domain name resolution capability - external
domain name: 10000 qps, internal domain name: 40000 qps",
      "Parameters.flavor3.name" : 10000,
      "Parameters.flavor4.description" : "Concurrent domain name resolution capability - 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" : "les spécifications du plugin peuvent être associées aux spécifications du cluster. le
fuseau horaire du plug-in est le même que celui du noeud",
      "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" : "Supports association between add-on specifications and cluster specifications. The

```

```
time zone of the add-on is the same as that of the node.",
  "description": "CoreDNS is a DNS server that chains plug-ins and offers DNS resolution for
Kubernetes clusters."
},
"description": {
  "Parameters.custom.stub_domains": "A domain name server for a custom domain name in key-
value pair. The key is a suffix of DNS domain name, and the value is one or more DNS IP addresses, for
example, **acme.local -- 1.2.3.4,6.7.8.9**.",
  "Parameters.custom.upstream_nameservers": "Resolves all domain names except intra-cluster
service domain names and custom domain names. The value can be one or more DNS IP addresses, for
example, **\"8.8.8.8\\\"\\\"8.8.4.4\\\"** ",
  "Parameters.flavor1.description": "Concurrent domain name resolution capability - external
domain name: 2500 qps, internal domain name: 10,000 qps",
  "Parameters.flavor1.name": 2500,
  "Parameters.flavor2.description": "Concurrent domain name resolution capability - external
domain name: 5000 qps, internal domain name: 20000 qps",
  "Parameters.flavor2.name": 5000,
  "Parameters.flavor3.description": "Concurrent domain name resolution capability - external
domain name: 10000 qps, internal domain name: 40000 qps",
  "Parameters.flavor3.name": 10000,
  "Parameters.flavor4.description": "Concurrent domain name resolution capability - 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 DNS server"
}
}
},
"supportVersions": null,
"creationTimestamp": "2024-01-22T11:05:45Z",
"updateTimestamp": "2024-01-22T11:05:45Z"
}
}
}
```

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 RollbackAutopilotAddonInstanceSolution {

    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();
    RollbackAutopilotAddonInstanceRequest request = new RollbackAutopilotAddonInstanceRequest();
    request.withId("{id}");
    AddonInstanceRollbackRequest body = new AddonInstanceRollbackRequest();
    body.withClusterID("*****");
    request.withBody(body);
    try {
        RollbackAutopilotAddonInstanceResponse response =
client.rollbackAutopilotAddonInstance(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

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = RollbackAutopilotAddonInstanceRequest()
        request.id = "{id}"
        request.body = AddonInstanceRollbackRequest(
            cluster_id="*****"
        )
        response = client.rollback_autopilot_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.RollbackAutopilotAddonInstanceRequest{}
    request.Id = "{id}"
    request.Body = &model.AddonInstanceRollbackRequest{
        ClusterID: "*****",
    }
    response, err := client.RollbackAutopilotAddonInstance(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 add-on instance version is rolled back. |

Error Codes

See [Error Codes](#).

4.2.5 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 /autopilot/v3/addons/{id}

Table 4-187 Path Parameters

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|--------------------|
| id | Yes | String | Add-on instance ID |

Table 4-188 Query Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|--|
| cluster_id | No | String | Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI . |

Request Parameters

Table 4-189 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Response Parameters

Status code: 200

Table 4-190 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 DeleteAutopilotAddonInstanceSolution {

    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();
        DeleteAutopilotAddonInstanceRequest request = new DeleteAutopilotAddonInstanceRequest();
        request.withId("{id}");
        try {
            DeleteAutopilotAddonInstanceResponse response = client.deleteAutopilotAddonInstance(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

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = DeleteAutopilotAddonInstanceRequest()
        request.id = "{id}"
        response = client.delete_autopilot_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.DeleteAutopilotAddonInstanceRequest{}
request.Id = "{id}"
response, err := client.DeleteAutopilotAddonInstance(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](#).

4.2.6 Obtaining 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 /autopilot/v3/addons/{id}

Table 4-191 Path Parameters

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|--------------------|
| id | Yes | String | Add-on instance ID |

Table 4-192 Query Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|--|
| cluster_id | No | String | Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI . |

Request Parameters

Table 4-193 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Response Parameters

Status code: 200

Table 4-194 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 4-195 AddonMetadata

| Parameter | Type | Description |
|-----------|--------|-------------|
| uid | String | Unique ID |
| name | String | Add-on name |

| Parameter | Type | Description |
|-------------------|--------------------|--|
| 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 4-196 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 4-197 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. |

| Parameter | Type | Description |
|-----------------|------------------------|--|
| 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. |
| previousVersion | String | The add-on version before upgrade or rollback |

Table 4-198 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 4-199 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. |

| Parameter | Type | Description |
|-----------|------------------|---|
| category | Array of strings | <p>The cluster types to which the settings are applied</p> <p>Options:</p> <ul style="list-style-type: none"> • CCE: CCE standard clusters • Turbo: CCE Turbo clusters • Autopilot: CCE Autopilot clusters <p>Default value:</p> <p>If this parameter is left blank, CCE standard and Turbo clusters are used by default.</p> |

Example Requests

None

Example Responses

Status code: 200

OK

```
{
  "kind": "Addon",
  "apiVersion": "v3",
  "metadata": {
    "uid": "90b775e0-5774-4e1d-ab3b-516332ba047a",
    "name": "coredns",
    "alias": "coredns",
    "creationTimestamp": "2024-07-18T04:04:21Z",
    "updateTimestamp": "2024-07-18T04:04:21Z"
  },
  "spec": {
    "clusterID": "597f2d95-44ab-11ef-9e39-0255ac100115",
    "version": "1.28.6",
    "addonTemplateName": "coredns",
    "addonTemplateType": "helm",
    "addonTemplateLogo": "",
    "addonTemplateLabels": [ "ContainerNetwork" ],
    "description": "CoreDNS is a DNS server that chains plugins and provides Kubernetes DNS Services",
    "values": null
  },
  "status": {
    "status": "abnormal",
    "Reason": "",
    "message": "",
    "targetVersions": null,
    "isRollbackable": false,
    "currentVersion": {
      "version": "1.28.6",
      "input": {
        "basic": {
          "cluster_ip": "10.247.3.10",
          "image_version": "1.28.6",
          "swr_addr": "swr.cn-north-7.myhuaweicloud.com",
          "swr_user": "autopilot-official"
        }
      }
    },
    "parameters": {
```



```

"autopilot-flavor1" : {
  "category" : [ "Autopilot" ],
  "is_default" : true,
  "name" : "autopilot-flavor1",
  "replicas" : 2,
  "resources" : [ {
    "limitsCpu" : 1,
    "limitsMem" : "2Gi",
    "name" : "coredns",
    "requestsCpu" : 1,
    "requestsMem" : "2Gi"
  } ]
},
"custom" : {
  "multiAZBalance" : false,
  "multiAZEnabled" : false,
  "node_match_expressions" : [ ],
  "parameterSyncStrategy" : "ensureConsistent",
  "servers" : [ {
    "plugins" : [ {
      "name" : "bind",
      "parameters" : "${POD_IP}"
    }, {
      "configBlock" : "servfail 5s",
      "name" : "cache",
      "parameters" : 30
    }, {
      "name" : "errors"
    }, {
      "name" : "health",
      "parameters" : "${POD_IP}:8080"
    }, {
      "name" : "ready",
      "parameters" : "${POD_IP}:8081"
    }, {
      "configBlock" : "pods insecure\nfallthrough in-addr.arpa ip6.arpa",
      "name" : "kubernetes",
      "parameters" : "cluster.local in-addr.arpa ip6.arpa"
    }, {
      "name" : "loadbalance",
      "parameters" : "round_robin"
    }, {
      "name" : "prometheus",
      "parameters" : "${POD_IP}:9153"
    }, {
      "configBlock" : "policy random",
      "name" : "forward",
      "parameters" : ". /etc/resolv.conf"
    }, {
      "name" : "reload"
    }
  ]
},
"port" : 5353,
"zones" : [ {
  "zone" : "."
} ]
}],
"stub_domains" : { },
"tolerations" : [ {
  "effect" : "NoExecute",
  "key" : "node.kubernetes.io/not-ready",
  "operator" : "Exists",
  "tolerationSeconds" : 60
}, {
  "effect" : "NoExecute",
  "key" : "node.kubernetes.io/unreachable",
  "operator" : "Exists",
  "tolerationSeconds" : 60
} ],
"upstream_nameservers" : [ ]

```

```

    },
    "flavor1" : {
      "is_default" : true,
      "name" : 2500,
      "recommend_cluster_flavor_types" : [ "small" ],
      "replicas" : 2,
      "resources" : [ {
        "limitsCpu" : "500m",
        "limitsMem" : "512Mi",
        "name" : "coredns",
        "requestsCpu" : "500m",
        "requestsMem" : "512Mi"
      } ]
    },
    "flavor2" : {
      "name" : 5000,
      "recommend_cluster_flavor_types" : [ "medium" ],
      "replicas" : 2,
      "resources" : [ {
        "limitsCpu" : "1000m",
        "limitsMem" : "1024Mi",
        "name" : "coredns",
        "requestsCpu" : "1000m",
        "requestsMem" : "1024Mi"
      } ]
    },
    "flavor3" : {
      "name" : 10000,
      "recommend_cluster_flavor_types" : [ "large" ],
      "replicas" : 2,
      "resources" : [ {
        "limitsCpu" : "2000m",
        "limitsMem" : "2048Mi",
        "name" : "coredns",
        "requestsCpu" : "2000m",
        "requestsMem" : "2048Mi"
      } ]
    },
    "flavor4" : {
      "name" : 20000,
      "recommend_cluster_flavor_types" : [ "xlarge" ],
      "replicas" : 4,
      "resources" : [ {
        "limitsCpu" : "2000m",
        "limitsMem" : "2048Mi",
        "name" : "coredns",
        "requestsCpu" : "2000m",
        "requestsMem" : "2048Mi"
      } ]
    }
  }
},
"stable" : true,
"translate" : {
  "en_US" : {
    "addon" : {
      "changeLog" : "Support autopilot cluster",
      "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 (for example, **\"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 capability - external domain name: 2500 qps, internal domain name: 10000 qps",

```

```

    "Parameters.flavor1.name" : 2500,
    "Parameters.flavor2.description" : "Concurrent domain name resolution capability - external
domain name: 5000 qps, internal domain name: 20000 qps",
    "Parameters.flavor2.name" : 5000,
    "Parameters.flavor3.description" : "Concurrent domain name resolution capability - external
domain name: 10000 qps, internal domain name: 40000 qps",
    "Parameters.flavor3.name" : 10000,
    "Parameters.flavor4.description" : "Concurrent domain name resolution capability - 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" : "les spécifications du plugin peuvent être associées aux spécifications du cluster. le
fuseau horaire du plug-in est le même que celui du noeud",
    "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" : "Autopilot clusters are supported.",
    "description" : "CoreDNS is a DNS server that chains plug-ins and offers DNS resolution for
Kubernetes clusters."
  },
  "description" : {
    "Parameters.custom.stub_domains" : "A domain name server for a custom domain name in key-
value pair. The key is a suffix of DNS domain name, and the value is one or more DNS IP addresses, for
example, **acme.local -- 1.2.3.4,6.7.8.9**.",
    "Parameters.custom.upstream_nameservers" : "Resolves all domain names except intra-cluster
service domain names and custom domain names. The value can be one or more DNS IP addresses, for
example, **\"8.8.8.8\"|\"8.8.4.4\"**.",
    "Parameters.flavor1.description" : "Concurrent domain name resolution capability - external
domain name: 2500 qps, internal domain name: 10000 qps",
    "Parameters.flavor1.name" : 2500,
    "Parameters.flavor2.description" : "Concurrent domain name resolution capability - external
domain name: 5000 qps, internal domain name: 20000 qps",
    "Parameters.flavor2.name" : 5000,
    "Parameters.flavor3.description" : "Concurrent domain name resolution capability - external
domain name: 10000 qps, internal domain name: 40000 qps",

```

```
    "Parameters.flavor3.name" : 10000,
    "Parameters.flavor4.description" : "Concurrent domain name resolution capability - 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 DNS server"
  }
},
"supportVersions" : null,
"creationTimestamp" : "2024-02-19T11:33:46Z",
"updateTimestamp" : "2024-02-21T01:24:05Z"
}
}
```

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 ShowAutopilotAddonInstanceSolution {

    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();
        ShowAutopilotAddonInstanceRequest request = new ShowAutopilotAddonInstanceRequest();
        request.withId("{id}");
        try {
            ShowAutopilotAddonInstanceResponse response = client.showAutopilotAddonInstance(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

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = ShowAutopilotAddonInstanceRequest()
        request.id = "{id}"
        response = client.show_autopilot_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.ShowAutopilotAddonInstanceRequest{}
request.Id = "{id}"
response, err := client.ShowAutopilotAddonInstance(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](#).

4.2.7 Listing Add-on Instances

Function

This API is used to obtain all add-on instances in a cluster.

Calling Method

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

URI

GET /autopilot/v3/addons

Table 4-200 Query 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 4-201 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Response Parameters

Status code: 200

Table 4-202 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 4-203 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 4-204 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 4-205 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 4-206 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. |

| Parameter | Type | Description |
|-----------------|------------------------|--|
| 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. |
| previousVersion | String | The add-on version before upgrade or rollback |

Table 4-207 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 4-208 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. |

| Parameter | Type | Description |
|-----------|------------------|---|
| category | Array of strings | <p>The cluster types to which the settings are applied</p> <p>Options:</p> <ul style="list-style-type: none"> • CCE: CCE standard clusters • Turbo: CCE Turbo clusters • Autopilot: CCE Autopilot clusters <p>Default value:</p> <p>If this parameter is left blank, CCE standard and Turbo clusters are used by default.</p> |

Example Requests

None

Example Responses

Status code: 200

ok

```
{
  "kind": "Addon",
  "apiVersion": "v3",
  "items": [ {
    "kind": "Addon",
    "apiVersion": "v3",
    "metadata": {
      "uid": "90b775e0-5774-4e1d-ab3b-516332ba047a",
      "name": "coredns",
      "alias": "coredns",
      "creationTimestamp": "2024-07-18T04:04:21Z",
      "updateTimestamp": "2024-07-18T04:04:21Z"
    },
    "spec": {
      "clusterID": "597f2d95-44ab-11ef-9e39-0255ac100115",
      "version": "1.28.6",
      "addonTemplateName": "coredns",
      "addonTemplateType": "helm",
      "addonTemplateLogo": "",
      "addonTemplateLabels": [ "ContainerNetwork" ],
      "description": "CoreDNS is a DNS server that chains plugins and provides Kubernetes DNS Services",
      "values": null
    },
    "status": {
      "status": "installing",
      "Reason": "",
      "message": "",
      "targetVersions": null,
      "isRollbackable": false,
      "currentVersion": {
        "version": "1.28.6",
        "input": {
          "basic": {
            "cluster_ip": "10.247.3.10",
            "image_version": "1.28.6",
            "swr_addr": "swr.cn-north-7.myhuaweicloud.com",
```

```

"swr_user" : "autopilot-official"
},
"parameters" : {
  "autopilot-flavor1" : {
    "category" : [ "Autopilot" ],
    "is_default" : true,
    "name" : "autopilot-flavor1",
    "replicas" : 2,
    "resources" : [ {
      "limitsCpu" : 1,
      "limitsMem" : "2Gi",
      "name" : "coredns",
      "requestsCpu" : 1,
      "requestsMem" : "2Gi"
    } ]
  }
},
"custom" : {
  "multiAZBalance" : false,
  "multiAZEnabled" : false,
  "node_match_expressions" : [ ],
  "parameterSyncStrategy" : "ensureConsistent",
  "servers" : [ {
    "plugins" : [ {
      "name" : "bind",
      "parameters" : "${POD_IP}"
    } ], {
      "configBlock" : "servfail 5s",
      "name" : "cache",
      "parameters" : 30
    }, {
      "name" : "errors"
    }, {
      "name" : "health",
      "parameters" : "${POD_IP}:8080"
    }, {
      "name" : "ready",
      "parameters" : "${POD_IP}:8081"
    }, {
      "configBlock" : "pods insecure\nfallthrough in-addr.arpa ip6.arpa",
      "name" : "kubernetes",
      "parameters" : "cluster.local in-addr.arpa ip6.arpa"
    }, {
      "name" : "loadbalance",
      "parameters" : "round_robin"
    }, {
      "name" : "prometheus",
      "parameters" : "${POD_IP}:9153"
    }, {
      "configBlock" : "policy random",
      "name" : "forward",
      "parameters" : ". /etc/resolv.conf"
    }, {
      "name" : "reload"
    }
  ],
  "port" : 5353,
  "zones" : [ {
    "zone" : ""
  } ]
},
"stub_domains" : { },
"tolerations" : [ {
  "effect" : "NoExecute",
  "key" : "node.kubernetes.io/not-ready",
  "operator" : "Exists",
  "tolerationSeconds" : 60
}, {
  "effect" : "NoExecute",
  "key" : "node.kubernetes.io/unreachable",
  "operator" : "Exists",

```

```

        "tolerationSeconds" : 60
    },
    "upstream_nameservers" : [ ]
},
"flavor1" : {
    "is_default" : true,
    "name" : 2500,
    "recommend_cluster_flavor_types" : [ "small" ],
    "replicas" : 2,
    "resources" : [ {
        "limitsCpu" : "500m",
        "limitsMem" : "512Mi",
        "name" : "coredns",
        "requestsCpu" : "500m",
        "requestsMem" : "512Mi"
    } ]
},
"flavor2" : {
    "name" : 5000,
    "recommend_cluster_flavor_types" : [ "medium" ],
    "replicas" : 2,
    "resources" : [ {
        "limitsCpu" : "1000m",
        "limitsMem" : "1024Mi",
        "name" : "coredns",
        "requestsCpu" : "1000m",
        "requestsMem" : "1024Mi"
    } ]
},
"flavor3" : {
    "name" : 10000,
    "recommend_cluster_flavor_types" : [ "large" ],
    "replicas" : 2,
    "resources" : [ {
        "limitsCpu" : "2000m",
        "limitsMem" : "2048Mi",
        "name" : "coredns",
        "requestsCpu" : "2000m",
        "requestsMem" : "2048Mi"
    } ]
},
"flavor4" : {
    "name" : 20000,
    "recommend_cluster_flavor_types" : [ "xlarge" ],
    "replicas" : 4,
    "resources" : [ {
        "limitsCpu" : "2000m",
        "limitsMem" : "2048Mi",
        "name" : "coredns",
        "requestsCpu" : "2000m",
        "requestsMem" : "2048Mi"
    } ]
}
},
"stable" : true,
"translate" : {
    "en_US" : {
        "addon" : {
            "changeLog" : "Support autopilot cluster",
            "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 (for example, **\"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 capability - external
domain name: 2500 qps, internal domain name: 10000 qps",
  "Parameters.flavor1.name" : 2500,
  "Parameters.flavor2.description" : "Concurrent domain name resolution capability - external
domain name: 5000 qps, internal domain name: 20000 qps",
  "Parameters.flavor2.name" : 5000,
  "Parameters.flavor3.description" : "Concurrent domain name resolution capability - external
domain name: 10000 qps, internal domain name: 40000 qps",
  "Parameters.flavor3.name" : 10000,
  "Parameters.flavor4.description" : "Concurrent domain name resolution capability - 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" : "les spécifications du plugin peuvent être associées aux spécifications du cluster. le
fuseau horaire du plug-in est le même que celui du noeud",
    "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" : "Autopilot clusters are supported.",
    "description" : "CoreDNS is a DNS server that chains plug-ins and offers DNS resolution for
Kubernetes clusters."
  },
  "description" : {
    "Parameters.custom.stub_domains" : "A domain name server for a custom domain name in key-
value pair. The key is a suffix of DNS domain name, and the value is one or more DNS IP addresses, for
example, **acme.local -- 1.2.3.4,6.7.8.9**.",
    "Parameters.custom.upstream_nameservers" : "Resolves all domain names except intra-cluster
service domain names and custom domain names. The value can be one or more DNS IP addresses, for
example, **\"8.8.8.8\", \"8.8.4.4\"**.",
    "Parameters.flavor1.description" : "Concurrent domain name resolution capability - external
domain name: 2500 qps, internal domain name: 10000 qps",
    "Parameters.flavor1.name" : 2500,
    "Parameters.flavor2.description" : "Concurrent domain name resolution capability - external

```

```
domain name: 5000 qps, internal domain name: 20000 qps",
  "Parameters.flavor2.name" : 5000,
  "Parameters.flavor3.description" : "Concurrent domain name resolution capability - external
domain name: 10000 qps, internal domain name: 40000 qps",
  "Parameters.flavor3.name" : 10000,
  "Parameters.flavor4.description" : "Concurrent domain name resolution capability - 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 DNS server"
}
},
"supportVersions" : null,
"creationTimestamp" : "2024-02-19T11:33:46Z",
"updateTimestamp" : "2024-02-21T01:24:05Z"
}
}
}
}
```

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 ListAutopilotAddonInstancesSolution {

    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();
        ListAutopilotAddonInstancesRequest request = new ListAutopilotAddonInstancesRequest();
        try {
            ListAutopilotAddonInstancesResponse response = client.listAutopilotAddonInstances(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

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = ListAutopilotAddonInstancesRequest()
        response = client.list_autopilot_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.ListAutopilotAddonInstancesRequest{}
response, err := client.ListAutopilotAddonInstances(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 Autopilot Cluster Upgrade

4.3.1 Upgrading a Cluster

Function

This API is used to upgrade a cluster.

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.
- Cluster upgrade APIs are available only upon request.

Calling Method

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

URI

POST /autopilot/v3/projects/{project_id}/clusters/{cluster_id}/operation/upgrade

Table 4-209 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| project_id | Yes | String | <p>Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Project IDs of the account</p> <p>Default value: N/A</p> |
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |

Request Parameters

Table 4-210 Request body parameters

| Parameter | Mandatory | Type | Description |
|-----------|-----------|---|---|
| metadata | Yes | <code>UpgradeClusterRequestMetadata</code> object | <p>Details: Cluster upgrade metadata information</p> <p>Constraints: None</p> |

| Parameter | Mandatory | Type | Description |
|-----------|-----------|---|--|
| spec | Yes | AutopilotUpgradeSpec object | <p>Details: Upgrade configuration information. CCE upgrades a cluster based on spec.</p> <p>Constraints: None</p> |

Table 4-211 UpgradeClusterRequestMetadata

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| apiVersion | Yes | String | <p>Details: API version</p> <p>Constraints: The value is fixed.</p> <p>Options:</p> <ul style="list-style-type: none"> v3 |
| kind | Yes | String | <p>Details: API type</p> <p>Constraints: The value is fixed.</p> <p>Options:</p> <ul style="list-style-type: none"> UpgradeTask |

Table 4-212 AutopilotUpgradeSpec

| Parameter | Mandatory | Type | Description |
|----------------------|-----------|--|---|
| clusterUpgradeAction | No | AutopilotClusterUpgradeAction object | <p>Details: Detailed configuration information about the cluster upgrade</p> <p>Constraints: None</p> |

Table 4-213 AutopilotClusterUpgradeAction

| Parameter | Mandatory | Type | Description |
|---------------|-----------|--|--|
| addons | No | Array of AutopilotUpgradeAddonConfig objects | <p>Details: Add-on configuration list. CCE upgrades add-ons based on the configuration during the cluster upgrade.</p> <p>Constraints: None</p> |
| nodeOrder | No | Map<String,Array< NodePriority >> | <p>Details: Upgrade sequence of nodes in a node pool. key indicates the node pool ID. The value for the default node pool is DefaultPool.</p> <p>Constraints: None</p> |
| nodePoolOrder | No | Map<String,Integer> | <p>Details: Upgrade sequence of a node pool, in key-value pairs. key indicates the node pool ID. The value for the default node pool is DefaultPool. value indicates the node pool priority. The default value is 0, which indicates the lowest priority. A larger value indicates a higher priority.</p> <p>Constraints: None</p> |
| strategy | Yes | UpgradeStrategy object | <p>Details: Cluster upgrade policy</p> <p>Constraints: None</p> |

| Parameter | Mandatory | Type | Description |
|---------------|-----------|--------|--|
| targetVersion | Yes | String | <p>Details: Target cluster version, for example, v1.23</p> <p>Constraints: You can only upgrade a cluster to a later version. Do not enter a value that is equal to or earlier than the current cluster version.</p> <p>Options: Supported cluster versions</p> |

Table 4-214 AutopilotUpgradeAddonConfig

| Parameter | Mandatory | Type | Description |
|-------------------|-----------|--------|--|
| addonTemplateName | Yes | String | <p>Details: CCE add-on name</p> <p>Constraints: None</p> <p>Options: Names of the add-ons installed in the cluster.</p> |
| operation | Yes | String | <p>Details: Action for upgrading an add-on</p> <p>Constraints: None</p> <p>Options: patch: specifies that the add-on version will be upgraded.</p> |
| version | Yes | String | <p>Details: Target add-on version</p> <p>Constraints: The target add-on version must match the target cluster version.</p> <p>Options: N/A</p> |

| Parameter | Mandatory | Type | Description |
|-----------|-----------|---------------------|---|
| values | No | Map<String, Object> | Details: Add-on parameter list, in key-value pairs Constraints: None |

Table 4-215 NodePriority

| Parameter | Mandatory | Type | Description |
|--------------|-----------|---------------------|---|
| nodeSelector | Yes | NodeSelector object | Details: Node label selector, which selects a batch of nodes Constraints: Only labels on the nodes are allowed. |
| priority | Yes | Integer | Details: Priority of the current batch of nodes. A larger value indicates a higher priority. Constraints: None Options: Positive integers Default value: 0 |

Table 4-216 NodeSelector

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|---|
| key | Yes | String | Details: Key Constraints: None Options: N/A |

| Parameter | Mandatory | Type | Description |
|-----------|-----------|------------------|---|
| value | No | Array of strings | Details: Tag value list Constraints: None Options: N/A |
| operator | Yes | String | Details: Logical operators of labels Constraints: None Options: <ul style="list-style-type: none"> • in • notin • exists • ! • gt • lt |

Table 4-217 UpgradeStrategy

| Parameter | Mandatory | Type | Description |
|----------------------|-----------|------------------------------------|--|
| type | Yes | String | Details: Upgrade policy type Constraints: None Options: <ul style="list-style-type: none"> • inPlaceRollingUpdate: in-place upgrade |
| inPlaceRollingUpdate | No | InPlaceRollingUpdate object | Details: Detailed configuration of an in-place upgrade policy Constraints: This parameter is mandatory when the in-place upgrade policy type is specified. |

Table 4-218 InPlaceRollingUpdate

| Parameter | Mandatory | Type | Description |
|-----------------|-----------|---------|--|
| userDefinedStep | No | Integer | <p>Details: Maximum number of nodes to be upgraded in a batch. Node pools will be upgraded in sequence. Nodes in node pools will be upgraded in batches. One node is upgraded in the first batch, two nodes in the second batch, and the number of nodes to be upgraded in each subsequent batch increases by a power of 2 until the maximum number of nodes to be upgraded in each batch is reached. The next cluster is upgraded after the previous one is upgraded.</p> <p>Constraints: None</p> <p>Options: 1 to 60</p> <p>Default value: 20</p> |
| scope | No | String | <p>Details: Scope of the node upgrade batch</p> <p>Constraints: None</p> <p>Options: Cluster: If the scope is set to a cluster, the upgrade batch will remain unchanged throughout the entire upgrade process. NodePool: If the scope is set to node pools, the upgrade batch will be reset for each individual node pool.</p> <p>Default value: Cluster</p> |

Response Parameters

Status code: 200

Table 4-219 Response body parameters

| Parameter | Type | Description |
|-----------|---|-----------------------|
| metadata | UpgradeClusterResponseMetadata object | Upgrade task metadata |
| spec | UpgradeResponseSpec object | Upgrade settings |

Table 4-220 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 4-221 UpgradeResponseSpec

| Parameter | Type | Description |
|----------------------|---|--------------------------|
| clusterUpgradeAction | ClusterUpgradeResponseAction object | Cluster upgrade settings |

Table 4-222 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 | UpgradeStrategyResponse object | Upgrade policies |
| config | Object | Cluster configuration specified during an upgrade |

Table 4-223 UpgradeStrategyResponse

| Parameter | Type | Description |
|----------------------|---|--|
| type | String | Upgrade policy type |
| inPlaceRollingUpdate | InPlaceRollingUpdateResponse object | Detailed configuration of an in-place upgrade policy |

Table 4-224 InPlaceRollingUpdateResponse

| Parameter | Type | Description |
|-----------------|---------|---|
| userDefinedStep | Integer | Maximum number of nodes to be upgraded in a batch |
| scope | String | Scope of the node upgrade batch |

Example Requests

Upgrade the cluster to v1.28 and set the node upgrade step to 20.

POST /autopilot/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 v1.28 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 UpgradeAutopilotClusterSolution {

    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();

        UpgradeAutopilotClusterRequest request = new UpgradeAutopilotClusterRequest();
        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 {
    UpgradeAutopilotClusterResponse response = client.upgradeAutopilotCluster(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

Upgrade the cluster to v1.28 and set the node upgrade step to 20.

```
# coding: utf-8

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["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 = UpgradeAutopilotClusterRequest()
        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_autopilot_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 v1.28 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.UpgradeAutopilotClusterRequest{
        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.UpgradeAutopilotCluster(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](#).

4.3.2 Obtaining Cluster Upgrade Task Details

Function

This API is used to obtain details about an upgrade task of a cluster. The task ID is 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.
- Cluster upgrade APIs are available only upon request.

Calling Method

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

URI

GET /autopilot/v3/projects/{project_id}/clusters/{cluster_id}/operation/upgrade/tasks/{task_id}

Table 4-225 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| project_id | Yes | String | <p>Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Project IDs of the account</p> <p>Default value: N/A</p> |
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |
| task_id | Yes | String | <p>Details: Upgrade task ID, obtained from the uid field in the response body after the cluster upgrade API is called.</p> <p>Constraints: None</p> <p>Options: Upgrade task IDs</p> <p>Default value: N/A</p> |

Request Parameters

None

Response Parameters

Status code: 200

Table 4-226 Response body parameters

| Parameter | Type | Description |
|------------|-----------------------------------|---|
| apiVersion | String | API version. Defaults to v3 . |
| kind | String | Resource type. Defaults to UpgradeTask . |
| metadata | UpgradeTaskMetadata object | Upgrade task metadata. |
| spec | UpgradeTaskSpec object | Upgrade task information. |
| status | UpgradeTaskStatus object | Upgrade task status. |

Table 4-227 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 4-228 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 4-229 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 cluster upgrade task details are 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 ShowAutopilotUpgradeClusterTaskSolution {

    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();

        ShowAutopilotUpgradeClusterTaskRequest request = new ShowAutopilotUpgradeClusterTaskRequest();
        request.withClusterId("{cluster_id}");
        request.withTaskId("{task_id}");
        try {
            ShowAutopilotUpgradeClusterTaskResponse response =
                client.showAutopilotUpgradeClusterTask(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

import os
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 = os.environ["CLOUD_SDK_AK"]
```

```
sk = os.environ["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 = ShowAutopilotUpgradeClusterTaskRequest()
    request.cluster_id = "{cluster_id}"
    request.task_id = "{task_id}"
    response = client.show_autopilot_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.ShowAutopilotUpgradeClusterTaskRequest{
        request.ClusterId = "{cluster_id}"
        request.TaskId = "{task_id}"
    }
    response, err := client.ShowAutopilotUpgradeClusterTask(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 upgrade task details are obtained. |

Error Codes

See [Error Codes](#).

4.3.3 Retrying a Cluster Upgrade Task

Function

This API is used to 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.
- Cluster upgrade APIs are available only upon request.

Calling Method

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

URI

POST /autopilot/v3/projects/{project_id}/clusters/{cluster_id}/operation/upgrade/retry

Table 4-230 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| project_id | Yes | String | <p>Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Project IDs of the account</p> <p>Default value: N/A</p> |
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |

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 RetryAutopilotUpgradeClusterTaskSolution {

    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();
        RetryAutopilotUpgradeClusterTaskRequest request = new RetryAutopilotUpgradeClusterTaskRequest();
        request.withClusterId("{cluster_id}");
        try {
            RetryAutopilotUpgradeClusterTaskResponse response =
                client.retryAutopilotUpgradeClusterTask(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

import os
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 = os.environ["CLOUD_SDK_AK"]
sk = os.environ["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 = RetryAutopilotUpgradeClusterTaskRequest()
    request.cluster_id = "{cluster_id}"
    response = client.retry_autopilot_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.RetryAutopilotUpgradeClusterTaskRequest{}
    request.ClusterId = "{cluster_id}"
    response, err := client.RetryAutopilotUpgradeClusterTask(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 upgrade retry task is delivered. |

Error Codes

See [Error Codes](#).

4.3.4 Obtaining a List of Cluster Upgrade Task Details

Function

This API is used to obtain a list of cluster upgrade task details.

Calling Method

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

URI

GET /autopilot/v3/projects/{project_id}/clusters/{cluster_id}/operation/upgrade/tasks

Table 4-231 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| project_id | Yes | String | <p>Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Project IDs of the account</p> <p>Default value: N/A</p> |

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|--|
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |

Request Parameters

None

Response Parameters

Status code: 200

Table 4-232 Response body parameters

| Parameter | Type | Description |
|------------|--|--|
| apiVersion | String | API version, which defaults to v3 |
| kind | String | Resource type |
| metadata | UpgradeTaskMetadata object | Metadata |
| items | Array of UpgradeTaskResponseBody objects | Cluster upgrade tasks |

Table 4-233 UpgradeTaskResponseBody

| Parameter | Type | Description |
|------------|--------|---|
| apiVersion | String | API version. Defaults to v3 . |
| kind | String | Resource type. Defaults to UpgradeTask . |

| Parameter | Type | Description |
|-----------|--|---------------------------|
| metadata | UpgradeTaskMetadata object | Upgrade task metadata. |
| spec | UpgradeTaskSpec object | Upgrade task information. |
| status | UpgradeTaskStatus object | Upgrade task status. |

Table 4-234 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 4-235 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 4-236 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 |

| Parameter | Type | Description |
|----------------|--------|-------------------------------|
| 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 ListAutopilotUpgradeClusterTasksSolution {

    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();
        ListAutopilotUpgradeClusterTasksRequest request = new ListAutopilotUpgradeClusterTasksRequest();
        request.withClusterId("{cluster_id}");
        try {
            ListAutopilotUpgradeClusterTasksResponse response =
client.listAutopilotUpgradeClusterTasks(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

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["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 = ListAutopilotUpgradeClusterTasksRequest()
    request.cluster_id = "{cluster_id}"
    response = client.list_autopilot_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.ListAutopilotUpgradeClusterTasksRequest{}
    request.ClusterId = "{cluster_id}"
    response, err := client.ListAutopilotUpgradeClusterTasks(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](#).

4.3.5 Performing a Pre-upgrade Check for a Cluster

Function

This API is used to perform a pre-upgrade check for a cluster.

Calling Method

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

URI

POST /autopilot/v3/projects/{project_id}/clusters/{cluster_id}/operation/precheck

Table 4-237 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| project_id | Yes | String | <p>Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Project IDs of the account</p> <p>Default value: N/A</p> |

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|--|
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |

Request Parameters

Table 4-238 Request body parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|-------------------------------------|---|
| apiVersion | Yes | String | <p>Details: API version</p> <p>Constraints: The value is fixed.</p> <p>Options:</p> <ul style="list-style-type: none"> v3 |
| kind | Yes | String | <p>Details: API type</p> <p>Constraints: The value is fixed.</p> <p>Options:</p> <ul style="list-style-type: none"> PreCheckTask |
| spec | Yes | PrecheckSpec object | <p>Details: 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 spec.</p> <p>Constraints: None</p> |

Table 4-239 PrecheckSpec

| Parameter | Mandatory | Type | Description |
|----------------------|-----------|--|---|
| clusterID | Yes | String | <p>Details: Cluster ID</p> <p>Constraints: None</p> <p>Options: N/A</p> |
| clusterVersion | Yes | String | <p>Details: Cluster version. The value is the patch version of the current cluster. You can log in to the console and view the version on the Overview page.</p> <p>Constraints: None</p> <p>Options: N/A</p> |
| targetVersion | Yes | String | <p>Details: Target version. If you enter a major version, the latest patch version is automatically selected.</p> <p>Constraints: None</p> <p>Options: Available cluster versions later than the current cluster version</p> |
| skippedCheckItemList | No | Array of skippedCheckItemList objects | <p>Details: Skipped check items</p> <p>Constraints: None</p> <p>Options: N/A</p> |

Table 4-240 skippedCheckItemList

| Parameter | Mandatory | Type | Description |
|------------------|-----------|-------------------------|--|
| name | No | String | <p>Details: Name of the skipped checked item</p> <p>Constraints: None</p> <p>Options: N/A</p> |
| resourceSelector | No | resourceSelector object | <p>Details: Resource tag selector</p> <p>Constraints: This parameter is available only for node check, but not for cluster check or add-on check.</p> <p>Options: N/A</p> |

Table 4-241 resourceSelector

| Parameter | Mandatory | Type | Description |
|-----------|-----------|------------------|---|
| key | Yes | String | <p>Details: Tag key</p> <p>Constraints: None</p> <p>Options: <ul style="list-style-type: none"> • node.uid: node UID </p> |
| values | No | Array of strings | <p>Details: Tag value list</p> <p>Constraints: None</p> <p>Options: N/A</p> |

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|---|
| operator | Yes | String | Details: Logical operators of labels Constraints: None Options: <ul style="list-style-type: none"> In |

Response Parameters

Status code: 200

Table 4-242 Response body parameters

| Parameter | Type | Description |
|------------|---|--|
| apiVersion | String | API version |
| kind | String | Resource type |
| metadata | PrecheckCluserResponseMetadata object | Pre-upgrade check metadata |
| spec | PrecheckCluserResponseSpec 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 4-243 PrecheckCluserResponseMetadata

| Parameter | Type | Description |
|-----------|--------|--------------------|
| uid | String | Check the task ID. |

Table 4-244 PrecheckCluserResponseSpec

| Parameter | Type | Description |
|-----------|--------|-------------|
| clusterID | String | Cluster ID |

| Parameter | Type | Description |
|--------------------------|---|-------------------------|
| clusterVersion | String | Current cluster version |
| targetVersion | String | Target version |
| skippedCheckItem List | Array of skippedCheckItemResponse objects | Skipped check items |

Table 4-245 skippedCheckItemResponse

| Parameter | Type | Description |
|------------------|---|--|
| name | String | Name of the item that is skipped to be checked |
| resourceSelector | resourceSelectorResponse object | Resource tag selector. This parameter is available only for node check, but not for cluster check or add-on check. |

Table 4-246 resourceSelectorResponse

| Parameter | Type | Description |
|-----------|------------------|----------------|
| key | String | Tag key |
| values | Array of strings | Tag value list |
| operator | String | Tag value |

Table 4-247 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 |

| Parameter | Type | Description |
|--------------------|---|--|
| 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 4-248 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 4-249 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 4-250 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 4-251 NodeStageStatus

| Parameter | Type | Description |
|-------------|--|------------------------------|
| nodeInfo | NodeInfo object | Node information |
| itemsStatus | Array of PreCheckItemStatus objects | Compliance set by check item |

Table 4-252 NodeInfo

| Parameter | Type | Description |
|-----------|--------|-------------|
| uid | String | Node UID |
| name | String | Node name |
| status | String | Status |
| nodeType | String | Node type |

Table 4-253 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 |

| Parameter | Type | Description |
|------------|-----------------------------------|--|
| 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 4-254 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 4-255 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 4-256 deprecatedAPIRisks

| Parameter | Type | Description |
|-----------|--------|--|
| url | String | Request path, for example, /apis/policy/v1beta1/podsecuritypolicies |
| userAgent | String | Client information |

Table 4-257 nodeRisks

| Parameter | Type | Description |
|-----------|--------|----------------|
| NodeID | String | Worker node ID |

Table 4-258 addonRisks

| Parameter | Type | Description |
|-------------------|--------|-----------------------------|
| addonTemplateName | String | Name of the add-on template |
| alias | String | Add-on alias |

Example Requests

Perform a pre-upgrade check for a cluster.

```
POST /autopilot/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"
    }
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Perform a pre-upgrade check for 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.List;
import java.util.ArrayList;

public class CreateAutopilotPreCheckSolution {

    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();

CreateAutopilotPreCheckRequest request = new CreateAutopilotPreCheckRequest();
request.withClusterId("{cluster_id}");
PrecheckClusterRequestBody body = new PrecheckClusterRequestBody();
PrecheckSpec specbody = new PrecheckSpec();
specbody.withClusterID("8978deaa-1743-11ee-8e46-0255ac10004c")
    .withClusterVersion("v1.15.11-r1")
    .withTargetVersion("v1.19.16-r80");
body.withSpec(specbody);
body.withKind("PreCheckTask");
body.withApiVersion("v3");
request.withBody(body);
try {
    CreateAutopilotPreCheckResponse response = client.createAutopilotPreCheck(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

Perform a pre-upgrade check for a cluster.

```
# coding: utf-8

import os
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 = os.getenv("CLOUD_SDK_AK")
    sk = 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 = CreateAutopilotPreCheckRequest()  
    request.cluster_id = "{cluster_id}"  
    specbody = PrecheckSpec(  
        cluster_id="8978deaa-1743-11ee-8e46-0255ac10004c",  
        cluster_version="v1.15.11-r1",  
        target_version="v1.19.16-r80"  
    )  
    request.body = PrecheckClusterRequestBody(  
        spec=specbody,  
        kind="PreCheckTask",  
        api_version="v3"  
    )  
    response = client.create_autopilot_pre_check(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

Perform a pre-upgrade check for 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.CreateAutopilotPreCheckRequest{  
        request.ClusterId = "{cluster_id}"  
        clusterIDSpec:= "8978deaa-1743-11ee-8e46-0255ac10004c"  
        clusterVersionSpec:= "v1.15.11-r1"  
        targetVersionSpec:= "v1.19.16-r80"  
        specbody := &model.PrecheckSpec{  
            ClusterID: &clusterIDSpec,  
            ClusterVersion: &clusterVersionSpec,
```

```

        TargetVersion: &targetVersionSpec,
    }
    request.Body = &model.PrecheckClusterRequestBody{
        Spec: specbody,
        Kind: "PreCheckTask",
        ApiVersion: "v3",
    }
    response, err := client.CreateAutopilotPreCheck(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 pre-upgrade check succeeded. |

Error Codes

See [Error Codes](#).

4.3.6 Obtaining Details About a Pre-upgrade Check Task of a Cluster

Function

This API is used to 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 /autopilot/v3/projects/{project_id}/clusters/{cluster_id}/operation/precheck/tasks/{task_id}

Table 4-259 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| project_id | Yes | String | <p>Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Project IDs of the account</p> <p>Default value: N/A</p> |
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |
| task_id | Yes | String | <p>Details: Upgrade task ID, obtained from the uid field in the response body after the cluster upgrade API is called.</p> <p>Constraints: None</p> <p>Options: Upgrade task IDs</p> <p>Default value: N/A</p> |

Request Parameters

None

Response Parameters

Status code: 200

Table 4-260 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 | PrecheckCluserResponseSpec object | Information of a pre-upgrade check task |
| status | PrecheckStatus object | Status of a pre-upgrade check task |

Table 4-261 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 4-262 PrecheckCluserResponseSpec

| Parameter | Type | Description |
|----------------------|--|-------------------------|
| clusterID | String | Cluster ID |
| clusterVersion | String | Current cluster version |
| targetVersion | String | Target version |
| skippedCheckItemList | Array of skippedCheckItemResponse objects | Skipped check items |

Table 4-263 skippedCheckItemListResponse

| Parameter | Type | Description |
|------------------|---|--|
| name | String | Name of the item that is skipped to be checked |
| resourceSelector | resourceSelectorResponse object | Resource tag selector. This parameter is available only for node check, but not for cluster check or add-on check. |

Table 4-264 resourceSelectorResponse

| Parameter | Type | Description |
|-----------|------------------|----------------|
| key | String | Tag key |
| values | Array of strings | Tag value list |
| operator | String | Tag value |

Table 4-265 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 4-266 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 4-267 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 4-268 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 4-269 NodeStageStatus

| Parameter | Type | Description |
|-------------|--|------------------------------|
| nodeInfo | NodeInfo object | Node information |
| itemsStatus | Array of PreCheckItemStatus objects | Compliance set by check item |

Table 4-270 NodeInfo

| Parameter | Type | Description |
|-----------|--------|-------------|
| uid | String | Node UID |
| name | String | Node name |
| status | String | Status |
| nodeType | String | Node type |

Table 4-271 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 4-272 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 4-273 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 4-274 deprecatedAPIRisks

| Parameter | Type | Description |
|-----------|--------|--|
| url | String | Request path, for example, /apis/policy/v1beta1/podsecuritypolicies |
| userAgent | String | Client information |

Table 4-275 nodeRisks

| Parameter | Type | Description |
|-----------|--------|----------------|
| NodeID | String | Worker node ID |

Table 4-276 addonRisks

| Parameter | Type | Description |
|-------------------|--------|-----------------------------|
| addonTemplateName | 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"
}
}
}
```

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 ShowAutopilotPreCheckSolution {
```

```
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();
    ShowAutopilotPreCheckRequest request = new ShowAutopilotPreCheckRequest();
    request.withClusterId("{cluster_id}");
    request.withTaskId("{task_id}");
    try {
        ShowAutopilotPreCheckResponse response = client.showAutopilotPreCheck(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

import os
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 = os.getenv("CLOUD_SDK_AK")
    sk = 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 = ShowAutopilotPreCheckRequest()
        request.cluster_id = "{cluster_id}"
```

```
request.task_id = "{task_id}"
response = client.show_autopilot_pre_check(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.ShowAutopilotPreCheckRequest{}
    request.ClusterId = "{cluster_id}"
    request.TaskId = "{task_id}"
    response, err := client.ShowAutopilotPreCheck(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 details about a pre-upgrade check task of a cluster are obtained. |

Error Codes

See [Error Codes](#).

4.3.7 Obtaining a List of Pre-upgrade Check Tasks of a Cluster

Function

This API is used to obtain a list of pre-upgrade check tasks of a cluster.

Calling Method

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

URI

GET /autopilot/v3/projects/{project_id}/clusters/{cluster_id}/operation/precheck/tasks

Table 4-277 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| project_id | Yes | String | <p>Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Project IDs of the account</p> <p>Default value: N/A</p> |

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|--|
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |

Request Parameters

None

Response Parameters

Status code: 200

Table 4-278 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. |
| items | Array of PrecheckCluster-Task objects | Cluster check tasks |

Table 4-279 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. |

| Parameter | Type | Description |
|-------------------|---------------------|--|
| annotations | Map<String,String > | Resource annotations in the format of key-value pairs. |
| updateTimestamp | String | Update time. |
| creationTimestamp | String | Creation time. |

Table 4-280 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 | PrecheckClusterResponseSpec object | Information of a pre-upgrade check task |
| status | PrecheckStatus object | Status of a pre-upgrade check task |

Table 4-281 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 4-282 PrecheckClusterResponseSpec

| Parameter | Type | Description |
|----------------|--------|-------------------------|
| clusterID | String | Cluster ID |
| clusterVersion | String | Current cluster version |
| targetVersion | String | Target version |

| Parameter | Type | Description |
|--------------------------|---|---------------------|
| skippedCheckItem List | Array of skippedCheckItemResponse objects | Skipped check items |

Table 4-283 skippedCheckItemListResponse

| Parameter | Type | Description |
|------------------|---|--|
| name | String | Name of the item that is skipped to be checked |
| resourceSelector | resourceSelectorResponse object | Resource tag selector. This parameter is available only for node check, but not for cluster check or add-on check. |

Table 4-284 resourceSelectorResponse

| Parameter | Type | Description |
|-----------|------------------|----------------|
| key | String | Tag key |
| values | Array of strings | Tag value list |
| operator | String | Tag value |

Table 4-285 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 |

| Parameter | Type | Description |
|------------------|---|---------------------------|
| addonCheckStatus | addonCheckStatus object | Status of an add-on check |
| nodeCheckStatus | nodeCheckStatus object | Node check status |

Table 4-286 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 4-287 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 4-288 nodeCheckStatus

| Parameter | Type | Description |
|-----------|--------|---|
| phase | String | Status. Options: <ul style="list-style-type: none"> • Init: initializing • Running • Success • Failed |

| Parameter | Type | Description |
|-----------------|---|-------------------|
| nodeStageStatus | Array of NodeStageStatus objects | Node check status |

Table 4-289 NodeStageStatus

| Parameter | Type | Description |
|-------------|--|------------------------------|
| nodeInfo | NodeInfo object | Node information |
| itemsStatus | Array of PreCheckItemStatus objects | Compliance set by check item |

Table 4-290 NodeInfo

| Parameter | Type | Description |
|-----------|--------|-------------|
| uid | String | Node UID |
| name | String | Node name |
| status | String | Status |
| nodeType | String | Node type |

Table 4-291 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 |

| Parameter | Type | Description |
|------------|-----------------------------------|---|
| 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 4-292 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 4-293 configurationRisks

| Parameter | Type | Description |
|------------|--------|------------------|
| package | String | Component |
| sourceFile | String | How to Obtain |
| nodeMsg | String | Node information |
| field | String | Parameter Value |

| Parameter | Type | Description |
|---------------|--------|---------------------------|
| operation | String | Change an operation type. |
| originalValue | String | Original Value |
| value | String | Current Value |

Table 4-294 deprecatedAPIRisks

| Parameter | Type | Description |
|-----------|--------|--|
| url | String | Request path, for example, /apis/policy/v1beta1/podsecuritypolicies |
| userAgent | String | Client information |

Table 4-295 nodeRisks

| Parameter | Type | Description |
|-----------|--------|----------------|
| NodeID | String | Worker node ID |

Table 4-296 addonRisks

| Parameter | Type | Description |
|-------------------|--------|-----------------------------|
| addonTemplateName | 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 ] status is abnormal, check and try again",
    "riskSource" : {
      "addonRisks" : [ {
        "addonTemplateName" : "coredns",
        "alias" : "CoreDNS"
      } ]
    }
  } ]
}

```

```
    }, {
      "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"
  }
}
}]
}
```

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 ListAutopilotPreCheckTasksSolution {

    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();
        ListAutopilotPreCheckTasksRequest request = new ListAutopilotPreCheckTasksRequest();
        request.withClusterId("{cluster_id}");
        try {
            ListAutopilotPreCheckTasksResponse response = client.listAutopilotPreCheckTasks(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

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["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 = ListAutopilotPreCheckTasksRequest()
        request.cluster_id = "{cluster_id}"
        response = client.list_autopilot_pre_check_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.ListAutopilotPreCheckTasksRequest{}
request.ClusterId = "{cluster_id}"
response, err := client.ListAutopilotPreCheckTasks(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 pre-upgrade check tasks of a cluster is obtained. |

Error Codes

See [Error Codes](#).

4.3.8 Performing a Post-upgrade Check for a Cluster

Function

This API is used to perform a post-upgrade check for a cluster, 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 /autopilot/v3/projects/{project_id}/clusters/{cluster_id}/operation/postcheck

Table 4-297 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| project_id | Yes | String | <p>Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Project IDs of the account</p> <p>Default value: N/A</p> |
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |

Request Parameters

Table 4-298 Request body parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| apiVersion | Yes | String | <p>Details: API version</p> <p>Constraints: The value is fixed.</p> <p>Options:</p> <ul style="list-style-type: none"> v3 |

| Parameter | Mandatory | Type | Description |
|-----------|-----------|-------------------------------------|---|
| kind | Yes | String | Details: API type Constraints: The value is fixed. Options: <ul style="list-style-type: none"> PostCheckTask |
| spec | Yes | PostcheckResponseSpec object | spec indicates the configuration confirmed after an upgrade. |

Table 4-299 PostcheckResponseSpec

| Parameter | Mandatory | Type | Description |
|----------------|-----------|--------|-------------------------|
| clusterID | No | String | Cluster ID |
| clusterVersion | No | String | Source cluster version |
| targetVersion | No | String | Current cluster version |

Response Parameters

Status code: 200

Table 4-300 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 4-301 PostcheckCluserResponseMetadata

| Parameter | Type | Description |
|-----------|--------|-------------|
| uid | String | Task ID |

Table 4-302 PostcheckSpec

| Parameter | Type | Description |
|----------------|--------|--|
| clusterID | String | Details: Cluster ID Constraints: None Options: N/A |
| clusterVersion | String | Details: Source cluster version Constraints: None Options: N/A |
| targetVersion | String | Details: Current cluster version Constraints: None Options: Supported cluster versions |

Table 4-303 status

| Parameter | Type | Description |
|-----------|--------|--|
| phase | String | Status. Options: <ul style="list-style-type: none"> • Success • Failed • Error |

Example Requests

Perform a post-upgrade check for a cluster.

```
POST /autopilot/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"
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Perform a post-upgrade check for 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 CreateAutopilotPostCheckSolution {

    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();
CreateAutopilotPostCheckRequest request = new CreateAutopilotPostCheckRequest();
request.withClusterId("{cluster_id}");
PostcheckClusterRequestBody body = new PostcheckClusterRequestBody();
PostcheckSpec specbody = new PostcheckSpec();
specbody.withClusterID("8978deaa-1743-11ee-8e46-0255ac10004c")
    .withClusterVersion("v1.15.11-r1")
    .withTargetVersion("v1.19.16-r80");
body.withSpec(specbody);
body.withKind("PostCheckTask");
body.withApiVersion("v3");
request.withBody(body);
try {
    CreateAutopilotPostCheckResponse response = client.createAutopilotPostCheck(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

Perform a post-upgrade check for a cluster.

```
# coding: utf-8

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["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 = CreateAutopilotPostCheckRequest()
        request.cluster_id = "{cluster_id}"
```

```
specbody = PostcheckSpec(  
    cluster_id="8978deaa-1743-11ee-8e46-0255ac10004c",  
    cluster_version="v1.15.11-r1",  
    target_version="v1.19.16-r80"  
)  
request.body = PostcheckClusterRequestBody(  
    spec=specbody,  
    kind="PostCheckTask",  
    api_version="v3"  
)  
response = client.create_autopilot_post_check(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

Perform a post-upgrade check for 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.CreateAutopilotPostCheckRequest{}  
    request.ClusterId = "{cluster_id}"  
    clusterIDSpec := "8978deaa-1743-11ee-8e46-0255ac10004c"  
    clusterVersionSpec := "v1.15.11-r1"  
    targetVersionSpec := "v1.19.16-r80"  
    specbody := &model.PostcheckSpec{  
        ClusterID: &clusterIDSpec,  
        ClusterVersion: &clusterVersionSpec,  
        TargetVersion: &targetVersionSpec,  
    }  
    request.Body = &model.PostcheckClusterRequestBody{  
        Spec: specbody,  
        Kind: "PostCheckTask",  
        ApiVersion: "v3",  
    }  
}
```



```
response, err := client.CreateAutopilotPostCheck(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 post-upgrade check succeeded. |

Error Codes

See [Error Codes](#).

4.3.9 Backing Up a Cluster

Function

This API is used to back up a cluster.

Calling Method

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

URI

POST /autopilot/v3.1/projects/{project_id}/clusters/{cluster_id}/operation/snapshot

Table 4-304 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| project_id | Yes | String | <p>Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Project IDs of the account</p> <p>Default value: N/A</p> |
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |

Request Parameters

None

Response Parameters

Status code: 200

Table 4-305 Response body parameters

| Parameter | Type | Description |
|-----------|--|-----------------|
| uid | String | Task ID |
| metadata | SnapshotClusterResponseMetadata object | Backup metadata |

Table 4-306 SnapshotClusterResponseMetadata

| Parameter | Type | Description |
|------------|--------|--|
| apiVersion | String | API version, which defaults to v3.1 |
| kind | String | Task type |

Example Requests

Back up a cluster before the cluster upgrade.

```
POST /autopilot/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"
  }
}
```

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 CreateAutopilotClusterMasterSnapshotSolution {

    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();
CreateAutopilotClusterMasterSnapshotRequest request = new
CreateAutopilotClusterMasterSnapshotRequest();
request.withClusterId("{cluster_id}");
try {
    CreateAutopilotClusterMasterSnapshotResponse response =
client.createAutopilotClusterMasterSnapshot(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

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["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 = CreateAutopilotClusterMasterSnapshotRequest()
        request.cluster_id = "{cluster_id}"
        response = client.create_autopilot_cluster_master_snapshot(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.CreateAutopilotClusterMasterSnapshotRequest{}
    request.ClusterId = "{cluster_id}"
    response, err := client.CreateAutopilotClusterMasterSnapshot(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 backup task is created. |

Error Codes

See [Error Codes](#).

4.3.10 Obtaining a List of Cluster Backup Task Details

Function

This API is used to obtain a list of cluster backup task details.

Calling Method

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

URI

GET /autopilot/v3.1/projects/{project_id}/clusters/{cluster_id}/operation/snapshot/tasks

Table 4-307 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| project_id | Yes | String | <p>Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Project IDs of the account</p> <p>Default value: N/A</p> |
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |

Request Parameters

None

Response Parameters

Status code: 200

Table 4-308 Response body parameters

| Parameter | Type | Description |
|------------|--------------------------------------|--|
| apiVersion | String | API version, which defaults to v3.1 |
| kind | String | Task type |
| metadata | SnapshotTaskMetadata object | Backup metadata |
| items | Array of SnapshotTask objects | Backup tasks |
| status | SnapshotTaskStatus object | Backup task status |

Table 4-309 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 4-310 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 4-311 SnapshotSpec

| Parameter | Type | Description |
|-----------|--|---------------------|
| items | Array of SnapshotSpecItems objects | Backup task details |

Table 4-312 SnapshotSpecItems

| Parameter | Type | Description |
|-------------------|--------|--------------------------------|
| id | String | Subtask ID |
| type | String | Subtask type |
| status | String | Status |
| creationTimestamp | String | Time when the task was created |
| updateTimestamp | String | Time when the task was updated |
| message | String | Information |

Table 4-313 SnapshotStatus

| Parameter | Type | Description |
|----------------|--------|-----------------|
| phase | String | Task status |
| progress | String | Task progress |
| completionTime | String | Completion time |

Table 4-314 SnapshotTaskStatus

| Parameter | Type | Description |
|------------------|--------|---------------------------|
| latestBackupTime | 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 ListAutopilotClusterMasterSnapshotTasksSolution {

    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();

        ListAutopilotClusterMasterSnapshotTasksRequest request = new
        ListAutopilotClusterMasterSnapshotTasksRequest();
        request.withClusterId("{cluster_id}");
        try {
            ListAutopilotClusterMasterSnapshotTasksResponse response =
```

```
client.listAutopilotClusterMasterSnapshotTasks(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

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["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 = ListAutopilotClusterMasterSnapshotTasksRequest()
        request.cluster_id = "{cluster_id}"
        response = client.list_autopilot_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.ListAutopilotClusterMasterSnapshotTasksRequest{}
request.ClusterId = "{cluster_id}"
response, err := client.ListAutopilotClusterMasterSnapshotTasks(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](#).

4.3.11 Obtaining the Cluster Upgrade Information

Function

This API is used to obtain the cluster upgrade information.

Calling Method

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

URI

GET /autopilot/v3/projects/{project_id}/clusters/{cluster_id}/upgradeinfo

Table 4-315 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| project_id | Yes | String | <p>Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Project IDs of the account</p> <p>Default value: N/A</p> |
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |

Request Parameters

None

Response Parameters

Status code: 200

Table 4-316 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. |

| Parameter | Type | Description |
|-----------|--|------------------|
| spec | UpgradeInfoSpec object | Upgrade settings |
| status | UpgradeInfoStatus object | Upgrade status |

Table 4-317 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 4-318 UpgradeInfoSpec

| Parameter | Type | Description |
|---------------------|--|-------------------------------|
| lastUpgradeInfo | UpgradeInfoStatus object | Last cluster upgrade |
| versionInfo | UpgradeVersionInfo object | Version |
| upgradeFeatureGates | UpgradeFeatureGates object | Cluster upgrade feature gates |

Table 4-319 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 |

| Parameter | Type | Description |
|----------------|------------------|---------------------------------|
| targetVersions | Array of strings | Target versions for the upgrade |

Table 4-320 UpgradeFeatureGates

| Parameter | Type | Description |
|----------------------|---------|---|
| supportUpgradePageV4 | Boolean | Whether the cluster upgrade console supports v4. This field is used by the CCE console. |

Table 4-321 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"
  }
}
```

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 ShowAutopilotClusterUpgradeInfoSolution {

    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();
        ShowAutopilotClusterUpgradeInfoRequest request = new ShowAutopilotClusterUpgradeInfoRequest();
        request.withClusterId("{cluster_id}");
        try {
            ShowAutopilotClusterUpgradeInfoResponse response =
                client.showAutopilotClusterUpgradeInfo(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  
  
import os  
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 = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["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 = ShowAutopilotClusterUpgradeInfoRequest()  
        request.cluster_id = "{cluster_id}"  
        response = client.show_autopilot_cluster_upgrade_info(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.ShowAutopilotClusterUpgradeInfoRequest{}  
request.ClusterId = "{cluster_id}"  
response, err := client.ShowAutopilotClusterUpgradeInfo(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 upgrade information is obtained. |

Error Codes

See [Error Codes](#).

4.3.12 Obtaining a Cluster Upgrade Path

Function

This API is used to obtain the cluster upgrade path.

Calling Method

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

URI

GET /autopilot/v3/clusterupgradepaths

Request Parameters

Table 4-322 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Response Parameters

Status code: 200

Table 4-323 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. |
| upgradePaths | Array of UpgradePath objects | Upgrade paths |

Table 4-324 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 4-325 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. |

| Parameter | Type | Description |
|-----------------|------------------|---|
| 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. |
| 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" ]
  } ]
}
```

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 ListAutopilotClusterUpgradePathsSolution {

    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();
        ListAutopilotClusterUpgradePathsRequest request = new ListAutopilotClusterUpgradePathsRequest();
        try {
            ListAutopilotClusterUpgradePathsResponse response =
client.listAutopilotClusterUpgradePaths(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

import os
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 = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]

credentials = BasicCredentials(ak, sk)

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

try:
    request = ListAutopilotClusterUpgradePathsRequest()
    response = client.list_autopilot_cluster_upgrade_paths(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.ListAutopilotClusterUpgradePathsRequest{}
    response, err := client.ListAutopilotClusterUpgradePaths(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 upgrade path is obtained. |

Error Codes

See [Error Codes](#).

4.3.13 Obtaining the Configuration of Cluster Upgrade Feature Gates

Function

This API is used to obtain the configuration of cluster upgrade feature gates.

Calling Method

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

URI

GET /autopilot/v3/clusterupgradefeaturegates

Request Parameters

Table 4-326 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Response Parameters

Status code: 200

Table 4-327 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. |
| upgradeFeatureGates | Map<String,String> | Feature gates information, which is in the format of a key-value pair. <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 4-328 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"
  }
}
```

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 ListAutopilotClusterUpgradeFeatureGatesSolution {

    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();

        ListAutopilotClusterUpgradeFeatureGatesRequest request = new
        ListAutopilotClusterUpgradeFeatureGatesRequest();
        try {
            ListAutopilotClusterUpgradeFeatureGatesResponse response =
            client.listAutopilotClusterUpgradeFeatureGates(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

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = ListAutopilotClusterUpgradeFeatureGatesRequest()
        response = client.list_autopilot_cluster_upgrade_feature_gates(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.ListAutopilotClusterUpgradeFeatureGatesRequest{}
response, err := client.ListAutopilotClusterUpgradeFeatureGates(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 upgrade path is obtained. |

Error Codes

See [Error Codes](#).

4.3.14 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 /autopilot/v3/projects/{project_id}/clusters/{cluster_id}/operation/upgradeworkflows

Table 4-329 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| project_id | Yes | String | <p>Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Project IDs of the account</p> <p>Default value: N/A</p> |
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |

Request Parameters

Table 4-330 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Table 4-331 Request body parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|-------------------------------------|--|
| kind | Yes | String | Details: API type Constraints: The value is fixed. Options: <ul style="list-style-type: none"> • WorkflowTask |
| apiVersion | Yes | String | Details: API version Constraints: The value is fixed. Options: <ul style="list-style-type: none"> • v3 |
| spec | Yes | WorkflowSpec object | Details: 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 . Constraints: None |

Table 4-332 WorkflowSpec

| Parameter | Mandatory | Type | Description |
|----------------|-----------|--------|---|
| clusterID | Yes | String | Details: Cluster ID Constraints: None Options: N/A |
| clusterVersion | No | String | Details: Current cluster version Constraints: None Options: N/A |

| Parameter | Mandatory | Type | Description |
|---------------|-----------|--------|--|
| targetVersion | Yes | String | Details: Target cluster version Constraints: None Options: N/A |

Response Parameters

Status code: 201

Table 4-333 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 | WorkflowResponseSpec 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 4-334 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. |

| Parameter | Type | Description |
|-------------------|---------------------|--|
| annotations | Map<String,String > | Resource annotations in the format of key-value pairs. |
| updateTimestamp | String | Update time. |
| creationTimestamp | String | Creation time. |

Table 4-335 WorkFlowResponseSpec

| Parameter | Type | Description |
|----------------|--------|-------------------------|
| clusterID | String | Cluster ID |
| clusterVersion | String | Current cluster version |
| targetVersion | String | Target cluster version |

Table 4-336 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 4-337 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 |
| expireTimeStamp | String | Expiration time of the upgrade task (only for pre-upgrade check tasks) |

Table 4-338 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 4-339 Point

| 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 |

Example Requests

Enable the process of upgrading a cluster to v1.28.

```
POST /autopilot/v3/projects/{project_id}/clusters/{cluster_id}/operation/upgradeworkflows
{
  "kind": "WorkFlowTask",
  "apiVersion": "v3",
  "spec": {
    "targetVersion": "v1.23",
    "clusterID": "7378a198-a3fe-11eb-ad37-0255ac100b07"
  }
}
```

Example Responses

Status code: 201

The upgrade process is created in the 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 v1.28.

```
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 CreateAutopilotUpgradeWorkFlowSolution {

    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();

        CreateAutopilotUpgradeWorkFlowRequest request = new CreateAutopilotUpgradeWorkFlowRequest();
        request.withClusterId("{cluster_id}");
        CreateUpgradeWorkFlowRequestBody body = new CreateUpgradeWorkFlowRequestBody();
        WorkFlowSpec specbody = new WorkFlowSpec();
        specbody.withClusterID("7378a198-a3fe-11eb-ad37-0255ac100b07")
            .withTargetVersion("v1.23");
        body.withSpec(specbody);
        body.withApiVersion("v3");
        body.withKind("WorkFlowTask");
        request.withBody(body);
        try {
            CreateAutopilotUpgradeWorkFlowResponse response =
```

```
client.createAutopilotUpgradeWorkFlow(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 v1.28.

```
# coding: utf-8

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["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 = CreateAutopilotUpgradeWorkFlowRequest()
        request.cluster_id = "{cluster_id}"
        specbody = WorkFlowSpec(
            cluster_id="7378a198-a3fe-11eb-ad37-0255ac100b07",
            target_version="v1.23"
        )
        request.body = CreateUpgradeWorkFlowRequestBody(
            spec=specbody,
            api_version="v3",
            kind="WorkFlowTask"
        )
        response = client.create_autopilot_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 v1.28.

```

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.CreateAutopilotUpgradeWorkFlowRequest{}
    request.ClusterId = "{cluster_id}"
    clusterIDSpec:= "7378a198-a3fe-11eb-ad37-0255ac100b07"
    specbody := &model.WorkFlowSpec{
        ClusterID: &clusterIDSpec,
        TargetVersion: "v1.23",
    }
    request.Body = &model.CreateUpgradeWorkFlowRequestBody{
        Spec: specbody,
        ApiVersion: "v3",
        Kind: "WorkFlowTask",
    }
    response, err := client.CreateAutopilotUpgradeWorkFlow(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 cluster. |

Error Codes

See [Error Codes](#).

4.3.15 Obtaining a List of Upgrade Workflows

Function

This API is used to obtain a list of historical cluster upgrade booting tasks.

Calling Method

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

URI

GET /autopilot/v3/projects/{project_id}/clusters/{cluster_id}/operation/upgradeworkflows

Table 4-340 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| project_id | Yes | String | <p>Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Project IDs of the account</p> <p>Default value: N/A</p> |
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |

Request Parameters

Table 4-341 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Response Parameters

Status code: 200

Table 4-342 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 | Upgrade workflow list |

Table 4-343 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 | WorkflowResponseSpec 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 4-344 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 4-345 WorkFlowResponseSpec

| Parameter | Type | Description |
|----------------|--------|-------------------------|
| clusterID | String | Cluster ID |
| clusterVersion | String | Current cluster version |
| targetVersion | String | Target cluster version |

Table 4-346 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 4-347 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 |
| expireTimeStamp | String | Expiration time of the upgrade task (only for pre-upgrade check tasks) |

Table 4-348 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 4-349 Point

| 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 |

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 ListAutopilotUpgradeWorkFlowsSolution {

    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();

        ListAutopilotUpgradeWorkFlowsRequest request = new ListAutopilotUpgradeWorkFlowsRequest();
        request.withClusterId("{cluster_id}");
        try {
            ListAutopilotUpgradeWorkFlowsResponse response =
                client.listAutopilotUpgradeWorkFlows(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

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["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 = ListAutopilotUpgradeWorkFlowsRequest()
        request.cluster_id = "{cluster_id}"
        response = client.list_autopilot_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.ListAutopilotUpgradeWorkFlowsRequest{}
        request.ClusterId = "{cluster_id}"
        response, err := client.ListAutopilotUpgradeWorkFlows(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](#).

4.3.16 Obtaining Details About a Specified Cluster Upgrade Booting 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 /autopilot/v3/projects/{project_id}/clusters/{cluster_id}/operation/upgradeworkflows/{upgrade_workflow_id}

Table 4-350 Path Parameters

| Parameter | Mandatory | Type | Description |
|---------------------|-----------|--------|---|
| project_id | Yes | String | <p>Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Project IDs of the account</p> <p>Default value: N/A</p> |
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |
| upgrade_workflow_id | Yes | String | <p>Booting process ID of a cluster upgrade task. For details about how to obtain the ID, see Obtaining API URI Parameters.</p> |

Request Parameters

Table 4-351 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Response Parameters

Status code: 200

Table 4-352 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 | WorkflowResponseSpec 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 4-353 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 4-354 WorkflowResponseSpec

| Parameter | Type | Description |
|----------------|--------|-------------------------|
| clusterID | String | Cluster ID |
| clusterVersion | String | Current cluster version |

| Parameter | Type | Description |
|---------------|--------|------------------------|
| targetVersion | String | Target cluster version |

Table 4-355 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 4-356 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 4-357 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 4-358 Point

| 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 |

Example Requests

None

Example Responses

Status code: 200

Details about the specified cluster upgrade booting 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 ShowAutopilotUpgradeWorkFlowSolution {

    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();
        ShowAutopilotUpgradeWorkFlowRequest request = new ShowAutopilotUpgradeWorkFlowRequest();
        request.withClusterId("{cluster_id}");
        request.withUpgradeWorkflowId("{upgrade_workflow_id}");
        try {
            ShowAutopilotUpgradeWorkFlowResponse response =
client.showAutopilotUpgradeWorkFlow(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

import os
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 = os.getenv("CLOUD_SDK_AK")
    sk = 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 = ShowAutopilotUpgradeWorkFlowRequest()
    request.cluster_id = "{cluster_id}"
    request.upgrade_workflow_id = "{upgrade_workflow_id}"
    response = client.show_autopilot_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.ShowAutopilotUpgradeWorkFlowRequest{}
    request.ClusterId = "{cluster_id}"
    request.UpgradeWorkflowId = "{upgrade_workflow_id}"
    response, err := client.ShowAutopilotUpgradeWorkFlow(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 cluster upgrade booting task are obtained. |

Error Codes

See [Error Codes](#).

4.3.17 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 /autopilot/v3/projects/{project_id}/clusters/{cluster_id}/operation/upgradeworkflows/{upgrade_workflow_id}

Table 4-359 Path Parameters

| Parameter | Mandatory | Type | Description |
|---------------------|-----------|--------|---|
| project_id | Yes | String | <p>Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Project IDs of the account</p> <p>Default value: N/A</p> |
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |
| upgrade_workflow_id | Yes | String | <p>Booting process ID of a cluster upgrade task. For details about how to obtain the ID, see Obtaining API URI Parameters.</p> |

Request Parameters

Table 4-360 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Table 4-361 Request body parameters

| Parameter | Mandatory | Type | Description |
|-----------|-----------|----------------------|--|
| status | No | status object | Details: Updated workflow status (Only Cancel is supported.) Constraints: None |

Table 4-362 status

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|---|
| phase | No | String | Details: Execution status of the cluster upgrade process Constraints: None Options: <ul style="list-style-type: none"> • Cancel: The upgrade is canceled. |

Response Parameters

Status code: 200

Table 4-363 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 | WorkflowResponseSpec 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 4-364 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 4-365 WorkFlowResponseSpec

| Parameter | Type | Description |
|----------------|--------|-------------------------|
| clusterID | String | Cluster ID |
| clusterVersion | String | Current cluster version |
| targetVersion | String | Target cluster version |

Table 4-366 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 4-367 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. |

| Parameter | Type | Description |
|-----------------|--------|--|
| 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 4-368 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 4-369 Point

| 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 |

Example Requests

Cancel the upgrade process.

```
PATCH /autopilot/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 UpgradeAutopilotWorkFlowUpdateSolution {

    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();

        UpgradeAutopilotWorkFlowUpdateRequest request = new
        UpgradeAutopilotWorkFlowUpdateRequest();
        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 {
            UpgradeAutopilotWorkFlowUpdateResponse response =
            client.upgradeAutopilotWorkFlowUpdate(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

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["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 = UpgradeAutopilotWorkFlowUpdateRequest()
        request.cluster_id = "{cluster_id}"
        request.upgrade_workflow_id = "{upgrade_workflow_id}"
        statusbody = UpgradeWorkFlowUpdateRequestBodyStatus(
            phase="Cancel"
        )
        request.body = UpgradeWorkFlowUpdateRequestBody(
            status=statusbody
        )
        response = client.upgrade_autopilot_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.UpgradeAutopilotWorkFlowUpdateRequest{}
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.UpgradeAutopilotWorkFlowUpdate(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](#).

4.4 Quota Management for Autopilot Clusters

4.4.1 Obtaining CCE Resource Quotas

Function

This API is used to get CCE resource quotas.

Calling Method

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

URI

GET /autopilot/v3/projects/{project_id}/quotas

Table 4-370 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| project_id | Yes | String | <p>Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Project IDs of the account</p> <p>Default value: N/A</p> |

Request Parameters

Table 4-371 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Response Parameters

Status code: 200

Table 4-372 Response body parameters

| Parameter | Type | Description |
|-----------|--|-------------|
| quotas | Array of QuotaResource objects | Resources |

Table 4-373 QuotaResource

| Parameter | Type | Description |
|------------|---------|--|
| quotaKey | String | <p>Details: Resource type</p> <p>Constraints: None</p> <p>Options:</p> <ul style="list-style-type: none"> cluster: cluster quota <p>Default value: N/A</p> |
| quotaLimit | Integer | Quota |

| Parameter | Type | Description |
|--------------------|---------|--|
| 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.

```
{
  "quotas": [ {
    "quotaKey": "autopilot_cluster",
    "quotaLimit": 20,
    "used": 13,
    "unit": "count",
    "regionId": "cn-north-7"
  } ]
}
```

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 ShowAutopilotQuotasSolution {

    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();
ShowAutopilotQuotasRequest request = new ShowAutopilotQuotasRequest();
try {
    ShowAutopilotQuotasResponse response = client.showAutopilotQuotas(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

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["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 = ShowAutopilotQuotasRequest()
        response = client.show_autopilot_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.ShowAutopilotQuotasRequest{}
    response, err := client.ShowAutopilotQuotas(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. |

Error Codes

See [Error Codes](#).

4.5 Tag Management for Autopilot Clusters

4.5.1 Adding Resource Tags to a Cluster in Batches

Function

This API is used to add resource tags to a 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 key and 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 /autopilot/v3/projects/{project_id}/clusters/{cluster_id}/tags/create

Table 4-374 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| project_id | Yes | String | <p>Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Project IDs of the account</p> <p>Default value: N/A</p> |
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |

Request Parameters

Table 4-375 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Table 4-376 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 4-377 ResourceTag

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|---|
| key | No | String | <p>Details: Key</p> <p>Constraints: None</p> <p>Options:</p> <ul style="list-style-type: none"> • The value cannot be empty and cannot start or end with spaces. A maximum of 128 characters are supported. • Letters, digits, and spaces in UTF-8 format are supported. • The value can contain the following special characters: <code>_:=-@</code> • The value cannot start with <code>_sys_</code>. <p>Default value: N/A</p> |

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|--|
| value | No | String | <p>Details: Value</p> <p>Constraints: None</p> <p>Options:</p> <ul style="list-style-type: none"> • The value can be null but not the default. Max characters: 255 • Letters, digits, and spaces in UTF-8 format are supported. • The value can contain the following special characters: _./=+-@ <p>Default value: N/A</p> |

Response Parameters

None

Example Requests

Add resource tags to a cluster in batches.

```
POST /autopilot/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

Add resource tags to a 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 BatchCreateAutopilotClusterTagsSolution {

    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();
        BatchCreateAutopilotClusterTagsRequest request = new BatchCreateAutopilotClusterTagsRequest();
        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 {
            BatchCreateAutopilotClusterTagsResponse response =
client.batchCreateAutopilotClusterTags(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 resource tags to a cluster in batches.

```
# coding: utf-8

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["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 = BatchCreateAutopilotClusterTagsRequest()
        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_autopilot_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

Add resource tags to a 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.BatchCreateAutopilotClusterTagsRequest{}
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.BatchCreateAutopilotClusterTags(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](#).

4.5.2 Deleting Resource Tags from a Cluster in Batches

Function

This API is used to delete resource tags from a cluster in batches.

 **NOTE**

- This API is idempotent. If the key of the to-be-deleted tag does not exist, the tag will be deleted.

Calling Method

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

URI

POST /autopilot/v3/projects/{project_id}/clusters/{cluster_id}/tags/delete

Table 4-378 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| project_id | Yes | String | <p>Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Project IDs of the account</p> <p>Default value: N/A</p> |
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |

Request Parameters

Table 4-379 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Table 4-380 Request body parameters

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--|---|
| tags | Yes | Array of ResourceDeleteTag objects | List of cluster resource tags to be deleted |

Table 4-381 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

Delete resource tags from a cluster in batches.

```
POST /autopilot/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

Delete resource tags from a 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 BatchDeleteAutopilotClusterTagsSolution {

    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();
        BatchDeleteAutopilotClusterTagsRequest request = new BatchDeleteAutopilotClusterTagsRequest();
        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 {
            BatchDeleteAutopilotClusterTagsResponse response =
            client.batchDeleteAutopilotClusterTags(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

Delete resource tags from a cluster in batches.

```
# coding: utf-8

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["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 = BatchDeleteAutopilotClusterTagsRequest()
        request.cluster_id = "{cluster_id}"
        listTagsbody = [
            ResourceDeleteTag(
                key="key1"
            ),
            ResourceDeleteTag(
                key="key2"
            )
        ]
        request.body = BatchDeleteClusterTagsRequestBody(
            tags=listTagsbody
        )
        response = client.batch_delete_autopilot_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

Delete resource tags from a 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.BatchDeleteAutopilotClusterTagsRequest{}
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.BatchDeleteAutopilotClusterTags(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](#).

4.6 Chart Management for Autopilot Clusters

4.6.1 Uploading a Chart

Function

This API is used to upload a chart.

Calling Method

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

URI

POST /autopilot/v2/charts

Request Parameters

Table 4-382 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Table 4-383 FormData parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| parameters | No | String | Configuration parameters for uploading a chart. The following shows an example: <code>"{"override":true,"skip_lint":true,"source":"package"}"</code> <ul style="list-style-type: none"> • skip_lint: indicates whether to verify the uploaded chart. • override: indicates whether to override an existing chart. • visible: indicates whether the chart is visible. |
| content | Yes | File | Chart package file |

Response Parameters

Status code: 201

Table 4-384 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 /autopilot/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-kubectrl\", \"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 UploadAutopilotChartSolution {

    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();
UploadAutopilotChartRequest request = new UploadAutopilotChartRequest();
UploadAutopilotChartRequestBody bodybody = new UploadAutopilotChartRequestBody();
bodybody.withParameters("{\"override\":true,\"skip_lint\":true,\"source\":\"package\"}")
    .withContent("chart-file.tgz");
body.withBody(bodybody);
request.withBody(listbodyBody);
try {
    UploadAutopilotChartResponse response = client.uploadAutopilotChart(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

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = UploadAutopilotChartRequest()
        bodybody = UploadAutopilotChartRequestBody(
            parameters={"override":true,"skip_lint":true,"source":"package"},
            content="chart-file.tgz"
        )
        request.body = listBodybody
        response = client.upload_autopilot_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.UploadAutopilotChartRequest{}
    parametersBody := "{\"override\":true,\"skip_lint\":true,\"source\":\"package\"}"
    bodybody := &model.UploadAutopilotChartRequestBody{
        Parameters: &parametersBody,
        Content: "chart-file.tgz",
    }
    request.Body = listBodybody
    response, err := client.UploadAutopilotChart(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](#).

4.6.2 Obtaining a Chart List

Function

This API is used to obtain a chart list.

Calling Method

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

URI

GET /autopilot/v2/charts

Request Parameters

Table 4-385 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Response Parameters

Status code: 200

Table 4-386 Response body parameters

| Parameter | Type | Description |
|-----------|--|-------------|
| [items] | Array of ChartResp objects | Chart list |

Table 4-387 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 |

| Parameter | Type | Description |
|-----------|---------|-----------------------------|
| 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" : "1abd3bd6-0258-11ec-b8b0-0255ac100b05",
  "name" : "magento-mysql",
  "values" : "",
  "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 ListAutopilotChartsSolution {

    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();
ListAutopilotChartsRequest request = new ListAutopilotChartsRequest();
try {
    ListAutopilotChartsResponse response = client.listAutopilotCharts(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

import os
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 = os.getenv("CLOUD_SDK_AK")
    sk = 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 = ListAutopilotChartsRequest()
        response = client.list_autopilot_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.ListAutopilotChartsRequest{}
    response, err := client.ListAutopilotCharts(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 Obtaining a Release List

Function

This API is used to obtain a release list.

Calling Method

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

URI

GET /autopilot/cam/v3/clusters/{cluster_id}/releases

Table 4-388 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|--|
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |

Table 4-389 Query Parameters

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|------------------------------------|
| chart_id | No | String | Chart ID |
| namespace | No | String | Namespace to which a chart belongs |

Request Parameters

Table 4-390 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Response Parameters

Status code: 200

Table 4-391 Response body parameters

| Parameter | Type | Description |
|-----------|--|-------------|
| [items] | Array of ReleaseResp objects | |

Table 4-392 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 |

| Parameter | Type | Description |
|--------------------|---------|---|
| status | String | Release status. <ul style="list-style-type: none"> • DEPLOYED: The release is normal. • DELETED: The release has been deleted. • FAILED: The release fails to be deployed. • DELETING: The release is being deleted. • PENDING_INSTALL: The release is waiting to be installed. • PENDING_UPGRADE: The release is waiting to be upgraded. • PENDING_ROLLBACK: The release is waiting for rollback. • UNKNOWN: The release status is unknown, indicating that the release is abnormal. You can manually delete the release and reinstall it. |
| 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":"csinas","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 ListAutopilotReleasesSolution {

    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();
        ListAutopilotReleasesRequest request = new ListAutopilotReleasesRequest();
        request.withClusterId("{cluster_id}");
        try {
            ListAutopilotReleasesResponse response = client.listAutopilotReleases(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

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = ListAutopilotReleasesRequest()
        request.cluster_id = "{cluster_id}"
        response = client.list_autopilot_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.ListAutopilotReleasesRequest{}
request.ClusterId = "{cluster_id}"
response, err := client.ListAutopilotReleases(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 Creating a Release

Function

This API is used to create a release.

Calling Method

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

URI

POST /autopilot/cam/v3/clusters/{cluster_id}/releases

Table 4-393 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|--|
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |

Request Parameters

Table 4-394 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Table 4-395 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 4-396 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 4-397 values

| Parameter | Mandatory | Type | Description |
|-----------------|-----------|--------|-------------------|
| imagePullPolicy | No | String | Image pull policy |
| imageTag | No | String | Image tag |

Response Parameters

Status code: 201

Table 4-398 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. <ul style="list-style-type: none"> • DEPLOYED: The release is normal. • DELETED: The release has been deleted. • FAILED: The release fails to be deployed. • DELETING: The release is being deleted. • PENDING_INSTALL: The release is waiting to be installed. • PENDING_UPGRADE: The release is waiting to be upgraded. • PENDING_ROLLBACK: The release is waiting for rollback. • UNKNOWN: The release status is unknown, indicating that the release is abnormal. You can manually delete the release and reinstall it. |
| status_description | String | Release status description |
| update_at | String | Updated at |
| values | String | Release value |
| version | Integer | Release version |

Example Requests

POST /autopilot/cam/v3/clusters/{cluster_id}/releases

```
{
  "name": "koi-neo4j",
  "namespace": "default",
  "chart_id": "e99a7e86-afdd-11eb-aca3-0255ac100b0e",
  "description": "",
  "version": "3.0.1",
  "values": {
    "imageTag": "v2",
    "imagePullPolicy": "IfNotPresent"
  }
}
```

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 CreateAutopilotReleaseSolution {

    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();
        CreateAutopilotReleaseRequest request = new CreateAutopilotReleaseRequest();
        request.withClusterId("{cluster_id}");
        CreateReleaseReqBody body = new CreateReleaseReqBody();
        CreateReleaseReqBodyValues valuesbody = new CreateReleaseReqBodyValues();
        valuesbody.withImagePullPolicy("IfNotPresent")
            .withImageTag("v2");
        body.withValues(valuesbody);
        body.withVersion("3.0.1");
        body.withNamespace("default");
        body.withName("koi-neo4j");
        body.withDescription("");
        body.withChartId("e99a7e86-afdd-11eb-aca3-0255ac100b0e");
        request.withBody(body);
        try {
            CreateAutopilotReleaseResponse response = client.createAutopilotRelease(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  
  
import os  
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 = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
  
    credentials = BasicCredentials(ak, sk)  
  
    client = CceClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(CceRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = CreateAutopilotReleaseRequest()  
        request.cluster_id = "{cluster_id}"  
        valuesbody = CreateReleaseReqBodyValues(  
            image_pull_policy="IfNotPresent",  
            image_tag="v2"  
        )  
        request.body = CreateReleaseReqBody(  
            values=valuesbody,  
            version="3.0.1",  
            namespace="default",  
            name="koi-neo4j",  
            description="",  
            chart_id="e99a7e86-afdd-11eb-aca3-0255ac100b0e"  
        )  
        response = client.create_autopilot_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.CreateAutopilotReleaseRequest{}
request.ClusterId = "{cluster_id}"
imagePullPolicyValues:= "IfNotPresent"
imageTagValues:= "v2"
valuesbody := &model.CreateReleaseReqBodyValues{
    ImagePullPolicy: &imagePullPolicyValues,
    ImageTag: &imageTagValues,
}
descriptionCreateReleaseReqBody:= ""
request.Body = &model.CreateReleaseReqBody{
    Values: valuesbody,
    Version: "3.0.1",
    Namespace: "default",
    Name: "koi-neo4j",
    Description: &descriptionCreateReleaseReqBody,
    ChartId: "e99a7e86-afdd-11eb-aca3-0255ac100b0e",
}
response, err := client.CreateAutopilotRelease(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](#).

4.6.5 Updating a Chart

Function

This API is used to update a chart.

Calling Method

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

URI

PUT /autopilot/v2/charts/{chart_id}

Table 4-399 Path Parameters

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|-------------|
| chart_id | Yes | String | Chart ID |

Request Parameters

Table 4-400 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Table 4-401 FormData parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|--|
| parameters | No | String | <p>Configuration parameters for updating a chart. The following shows an example: "{"override":true,"skip_lint":true,"source":"package"}"</p> <ul style="list-style-type: none"> • skip_lint: indicates whether to verify the uploaded chart. • override: indicates whether to override an existing chart. • visible: indicates whether the chart is visible. |
| content | Yes | File | Chart package file |

Response Parameters

Status code: 200

Table 4-402 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

```
PUT /autopilot/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" : "2021-05-08T08:53:12Z",  
"update_at" : "2021-05-08T08:53:12Z"  
}
```

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 UpdateAutopilotChartSolution {  
  
    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();  
        UpdateAutopilotChartRequest request = new UpdateAutopilotChartRequest();  
        request.withChartId("{chart_id}");  
        UpdateAutopilotChartRequestBody bodybody = new UpdateAutopilotChartRequestBody();  
        bodybody.withParameters("{\"override\":true,\"skip_lint\":true,\"source\":\"package\"}")  
            .withContent("chart-file.tgz");  
        body.withBody(bodybody);  
        request.withBody(listbodyBody);  
        try {  
            UpdateAutopilotChartResponse response = client.updateAutopilotChart(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

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = UpdateAutopilotChartRequest()
        request.chart_id = "{chart_id}"
        bodybody = UpdateAutopilotChartRequestBody(
            parameters="{\"override\":true,\"skip_lint\":true,\"source\":\"package\"}",
            content="chart-file.tgz"
        )
        request.body = listBodybody
        response = client.update_autopilot_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.UpdateAutopilotChartRequest{}
request.ChartId = "{chart_id}"
parametersBody:= "{\"override\":true,\"skip_lint\":true,\"source\":\"package\"}"
bodybody := &model.UpdateAutopilotChartRequestBody{
  Parameters: &parametersBody,
  Content: "chart-file.tgz",
}
request.Body = listBodybody
response, err := client.UpdateAutopilotChart(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 Deleting a Chart

Function

This API is used to delete a chart.

Calling Method

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

URI

DELETE /autopilot/v2/charts/{chart_id}

Table 4-403 Path Parameters

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|-------------|
| chart_id | Yes | String | Chart ID |

Request Parameters

Table 4-404 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Response Parameters

Status code: 200

Table 4-405 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 DeleteAutopilotChartSolution {

    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();
        DeleteAutopilotChartRequest request = new DeleteAutopilotChartRequest();
        request.withChartId("{chart_id}");
        try {
            DeleteAutopilotChartResponse response = client.deleteAutopilotChart(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

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = DeleteAutopilotChartRequest()
        request.chart_id = "{chart_id}"
        response = client.delete_autopilot_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.DeleteAutopilotChartRequest{}
request.ChartId = "{chart_id}"
response, err := client.DeleteAutopilotChart(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 Updating a Release

Function

This API is used to update a release.

Calling Method

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

URI

PUT /autopilot/cam/v3/clusters/{cluster_id}/namespace/{namespace}/releases/{name}

Table 4-406 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|--|
| name | Yes | String | Release name |
| namespace | Yes | String | Namespace to which a chart release belongs |
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |

Request Parameters

Table 4-407 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Table 4-408 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 4-409 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 |

| Parameter | Mandatory | Type | Description |
|-----------------|-----------|---------|---|
| 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 4-410 values

| Parameter | Mandatory | Type | Description |
|-----------------|-----------|--------|-------------------|
| imagePullPolicy | No | String | Image pull policy |
| imageTag | No | String | Image tag |

Response Parameters

Status code: 200

Table 4-411 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 |

| Parameter | Type | Description |
|--------------------|---------|---|
| resources | String | Resources required by the release |
| status | String | Release status. <ul style="list-style-type: none"> • DEPLOYED: The release is normal. • DELETED: The release has been deleted. • FAILED: The release fails to be deployed. • DELETING: The release is being deleted. • PENDING_INSTALL: The release is waiting to be installed. • PENDING_UPGRADE: The release is waiting to be upgraded. • PENDING_ROLLBACK: The release is waiting for rollback. • UNKNOWN: The release status is unknown, indicating that the release is abnormal. You can manually delete the release and reinstall it. |
| status_description | String | Release status description |
| update_at | String | Updated at |
| values | String | Release value |
| version | Integer | Release version |

Example Requests

```
PUT /autopilot/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 UpdateAutopilotReleaseSolution {

    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();
    }
}
```

```
UpdateAutopilotReleaseRequest request = new UpdateAutopilotReleaseRequest();
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 {
    UpdateAutopilotReleaseResponse response = client.updateAutopilotRelease(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

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = UpdateAutopilotReleaseRequest()
        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_autopilot_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.UpdateAutopilotReleaseRequest{}
    request.Name = "{name}"
    request.Namespace = "{namespace}"
    request.ClusterId = "{cluster_id}"
    request.ImagePullPolicyValues := "IfNotPresent"
    request.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.UpdateAutopilotRelease(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](#).

4.6.8 Obtaining a Chart

Function

This API is used to obtain a chart.

Calling Method

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

URI

GET /autopilot/v2/charts/{chart_id}

Table 4-412 Path Parameters

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|-------------|
| chart_id | Yes | String | Chart ID |

Request Parameters

Table 4-413 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Response Parameters

Status code: 200

Table 4-414 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" : "{ \"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 ShowAutopilotChartSolution {

    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();
ShowAutopilotChartRequest request = new ShowAutopilotChartRequest();
request.withChartId("{chart_id}");
try {
    ShowAutopilotChartResponse response = client.showAutopilotChart(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

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = ShowAutopilotChartRequest()
        request.chart_id = "{chart_id}"
        response = client.show_autopilot_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.ShowAutopilotChartRequest{}
    request.ChartId = "{chart_id}"
    response, err := client.ShowAutopilotChart(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](#).

4.6.9 Deleting a Release

Function

This API is used to delete a release.

Calling Method

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

URI

DELETE /autopilot/cam/v3/clusters/{cluster_id}/namespace/{namespace}/releases/{name}

Table 4-415 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|--|
| name | Yes | String | Release name |
| namespace | Yes | String | Namespace to which a chart release belongs |
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |

Request Parameters

Table 4-416 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Response Parameters

Status code: 200

Table 4-417 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 DeleteAutopilotReleaseSolution {

    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();
        DeleteAutopilotReleaseRequest request = new DeleteAutopilotReleaseRequest();
        request.withName("{name}");
        request.withNamespace("{namespace}");
        request.withClusterId("{cluster_id}");
        try {
            DeleteAutopilotReleaseResponse response = client.deleteAutopilotRelease(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

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = DeleteAutopilotReleaseRequest()
        request.name = "{name}"
        request.namespace = "{namespace}"
        request.cluster_id = "{cluster_id}"
        response = client.delete_autopilot_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.DeleteAutopilotReleaseRequest{}
request.Name = "{name}"
request.Namespace = "{namespace}"
request.ClusterId = "{cluster_id}"
response, err := client.DeleteAutopilotRelease(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](#).

4.6.10 Obtaining a Release

Function

This API is used to obtain a release.

Calling Method

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

URI

GET /autopilot/cam/v3/clusters/{cluster_id}/namespace/{namespace}/releases/{name}

Table 4-418 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|--|
| name | Yes | String | Release name |
| namespace | Yes | String | Namespace to which a chart release belongs |
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |

Request Parameters

Table 4-419 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Response Parameters

Status code: 200

Table 4-420 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 |

| Parameter | Type | Description |
|--------------------|---------|---|
| status | String | Release status. <ul style="list-style-type: none"> • DEPLOYED: The release is normal. • DELETED: The release has been deleted. • FAILED: The release fails to be deployed. • DELETING: The release is being deleted. • PENDING_INSTALL: The release is waiting to be installed. • PENDING_UPGRADE: The release is waiting to be upgraded. • PENDING_ROLLBACK: The release is waiting for rollback. • UNKNOWN: The release status is unknown, indicating that the release is abnormal. You can manually delete the release and reinstall it. |
| 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 ShowAutopilotReleaseSolution {

    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();
        ShowAutopilotReleaseRequest request = new ShowAutopilotReleaseRequest();
        request.setName("{name}");
        request.withNamespace("{namespace}");
        request.withClusterId("{cluster_id}");
        try {
            ShowAutopilotReleaseResponse response = client.showAutopilotRelease(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

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = ShowAutopilotReleaseRequest()
        request.name = "{name}"
        request.namespace = "{namespace}"
        request.cluster_id = "{cluster_id}"
        response = client.show_autopilot_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.ShowAutopilotReleaseRequest{}
request.Name = "{name}"
request.Namespace = "{namespace}"
request.ClusterId = "{cluster_id}"
response, err := client.ShowAutopilotRelease(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](#).

4.6.11 Downloading a Chart

Function

This API is used to download a chart.

Calling Method

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

URI

GET /autopilot/v2/charts/{chart_id}/archive

Table 4-421 Path Parameters

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|-------------|
| chart_id | Yes | String | Chart ID |

Request Parameters

Table 4-422 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Response Parameters

Status code: 200

Table 4-423 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 DownloadAutopilotChartSolution {

    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();
        DownloadAutopilotChartRequest request = new DownloadAutopilotChartRequest();
        request.withChartId("{chart_id}");
        try {
            DownloadAutopilotChartResponse response = client.downloadAutopilotChart(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

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = DownloadAutopilotChartRequest()
        request.chart_id = "{chart_id}"
        response = client.download_autopilot_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.DownloadAutopilotChartRequest{}
request.ChartId = "{chart_id}"
response, err := client.DownloadAutopilotChart(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](#).

4.6.12 Obtaining Chart Values

Function

This API is used to obtain chart values.

Calling Method

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

URI

GET /autopilot/v2/charts/{chart_id}/values

Table 4-424 Path Parameters

| Parameter | Mandatory | Type | Description |
|-----------|-----------|--------|-------------|
| chart_id | Yes | String | Chart ID |

Request Parameters

Table 4-425 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Response Parameters

Status code: 200

Table 4-426 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 ShowAutopilotChartValuesSolution {
    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();

ShowAutopilotChartValuesRequest request = new ShowAutopilotChartValuesRequest();
request.withChartId("{chart_id}");
try {
    ShowAutopilotChartValuesResponse response = client.showAutopilotChartValues(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

import os
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 = os.getenv("CLOUD_SDK_AK")
    sk = 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 = ShowAutopilotChartValuesRequest()
        request.chart_id = "{chart_id}"
        response = client.show_autopilot_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.ShowAutopilotChartValuesRequest{}
    request.ChartId = "{chart_id}"
    response, err := client.ShowAutopilotChartValues(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](#).

4.6.13 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 /autopilot/cam/v3/clusters/{cluster_id}/namespace/{namespace}/releases/{name}/history

Table 4-427 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|--|
| name | Yes | String | Release name |
| namespace | Yes | String | Namespace to which a chart release belongs |
| cluster_id | Yes | String | <p>Details: Cluster ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Cluster IDs</p> <p>Default value: N/A</p> |

Request Parameters

Table 4-428 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Response Parameters

Status code: 200

Table 4-429 Response body parameters

| Parameter | Type | Description |
|-----------|--|-------------|
| [items] | Array of ReleaseResp objects | |

Table 4-430 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 |

| Parameter | Type | Description |
|--------------------|---------|---|
| status | String | Release status. <ul style="list-style-type: none"> • DEPLOYED: The release is normal. • DELETED: The release has been deleted. • FAILED: The release fails to be deployed. • DELETING: The release is being deleted. • PENDING_INSTALL: The release is waiting to be installed. • PENDING_UPGRADE: The release is waiting to be upgraded. • PENDING_ROLLBACK: The release is waiting for rollback. • UNKNOWN: The release status is unknown, indicating that the release is abnormal. You can manually delete the release and reinstall it. |
| 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 ShowAutopilotReleaseHistorySolution {

    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();
        ShowAutopilotReleaseHistoryRequest request = new ShowAutopilotReleaseHistoryRequest();
        request.setName("{name}");
        request.withNamespace("{namespace}");
        request.withClusterId("{cluster_id}");
        try {
            ShowAutopilotReleaseHistoryResponse response = client.showAutopilotReleaseHistory(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

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = ShowAutopilotReleaseHistoryRequest()
        request.name = "{name}"
        request.namespace = "{namespace}"
        request.cluster_id = "{cluster_id}"
        response = client.show_autopilot_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.ShowAutopilotReleaseHistoryRequest{}  
request.Name = "{name}"  
request.Namespace = "{namespace}"  
request.ClusterId = "{cluster_id}"  
response, err := client.ShowAutopilotReleaseHistory(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](#).

4.6.14 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 /autopilot/v2/charts/{project_id}/quotas

Table 4-431 Path Parameters

| Parameter | Mandatory | Type | Description |
|------------|-----------|--------|---|
| project_id | Yes | String | <p>Details: Project ID. For details about how to obtain the value, see How to Obtain Parameters in the API URI.</p> <p>Constraints: None</p> <p>Options: Project IDs of the account</p> <p>Default value: N/A</p> |

Request Parameters

Table 4-432 Request header parameters

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|--|
| Content-Type | Yes | String | <p>Details: The request body type or format</p> <p>Constraints: The GET method is not verified.</p> <p>Options:</p> <ul style="list-style-type: none"> • application/json • application/json;charset=utf-8 • application/x-pem-file • multipart/form-data (used when the FormData parameter is present) <p>Default value: N/A</p> |

| Parameter | Mandatory | Type | Description |
|--------------|-----------|--------|---|
| X-Auth-Token | Yes | String | <p>Details: 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> <p>Constraints: None</p> <p>Options: N/A</p> <p>Default value: N/A</p> |

Response Parameters

Status code: 200

Table 4-433 Response body parameters

| Parameter | Type | Description |
|-----------|-------------------------------|-------------|
| quotas | quotas object | Chart quota |

Table 4-434 quotas

| Parameter | Type | Description |
|-----------|--|-------------|
| resources | Array of resources objects | Resources |

Table 4-435 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 ShowAutopilotUserChartsQuotasSolution {

    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();
        ShowAutopilotUserChartsQuotasRequest request = new ShowAutopilotUserChartsQuotasRequest();
        try {
            ShowAutopilotUserChartsQuotasResponse response =
            client.showAutopilotUserChartsQuotas(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

import os
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 = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["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 = ShowAutopilotUserChartsQuotasRequest()
        response = client.show_autopilot_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.ShowAutopilotUserChartsQuotasRequest{}
response, err := client.ShowAutopilotUserChartsQuotas(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](#).

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.

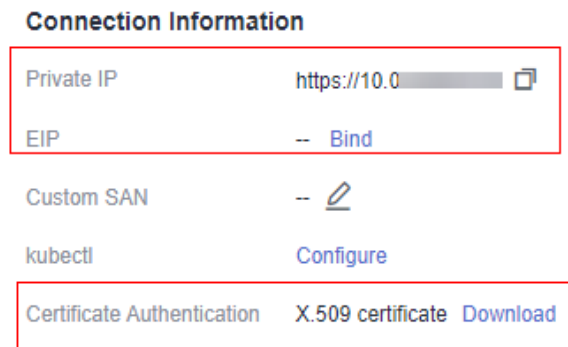
Calling Kubernetes APIs Through the Cluster API Server

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: Obtain the API server address (private or public network address) on the [Overview](#) page of the CCE console and download certificates (`client.crt` and `client.key`).



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, which 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 obtain the cluster UID.
- Method 2: Obtain the cluster on the **Overview** 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](#).
- You can obtain the parameter value 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 |

| Language | Client Library | Sample Program |
|------------|---|------------------------|
| 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 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 6-1 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 6-2 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 | √ | √ NOTE If you use enterprise project authorization to create a node, you need to add the global permission of evs:quota:get . |
| 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 6-3 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 DELETE /api/v2/projects/{project_id}/jobs/{job_id} | cce:job:delete | √ | √ |

Table 6-4 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 6-5 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 6-6 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 6-7 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 6-8 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 6-9 Quota management actions

| Permission | API | Action | IAM Project | Enterprise Project |
|-------------------------|--|---------------|-------------|--------------------|
| Obtaining quota details | GET /api/v3/projects/{project_id}/quotas | cce:quota:get | √ | √ |

7 Appendix

7.1 Status Code

[Table 7-1](#) describes the status codes.

Table 7-1 Status code

| Stat us Cod 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. 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. 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 us Cod 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 us Cod e | Code | Description |
|------------------------|-------------------------------|---|
| 403 | Forbidden | The server has received the request and understood it, but the server is refusing to respond to it. The client should modify the request instead of re-initiating it. |
| 404 | NotFound | The requested resource cannot be found. The client should modify the request instead of re-initiating it. |
| 405 | MethodNotAllowed | A request method is not supported for the requested resource. 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. The client may re-initiate the request without modifications at any later time. |
| 409 | Conflict | The request cannot be processed due to a conflict. 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. 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 us Cod 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. 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. |

7.2 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. |

7.3 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.

3. Hover over the username and choose **My Credentials** from the drop-down list.

On the **Projects** tab page, view project IDs.

Figure 7-1 Viewing project IDs

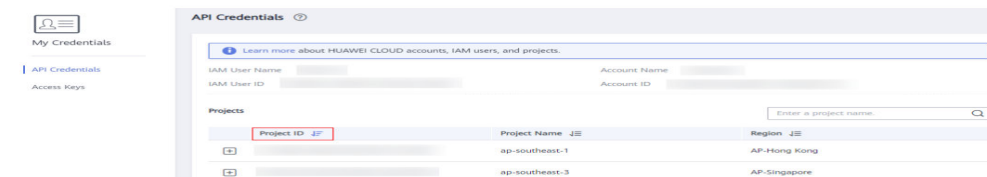
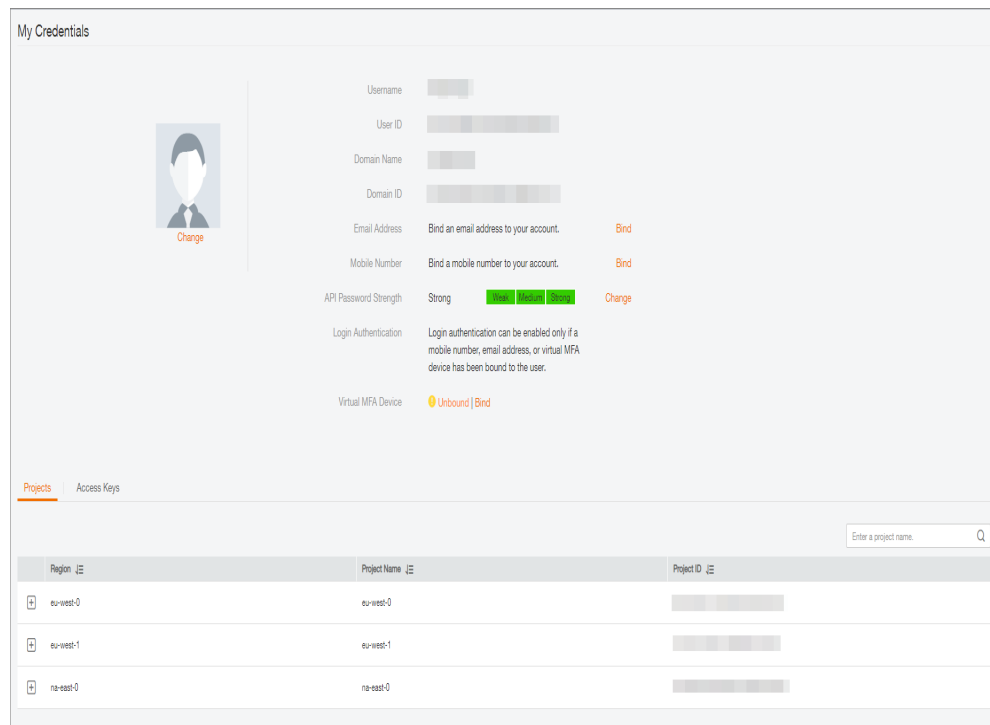


Figure 7-2 Viewing project IDs



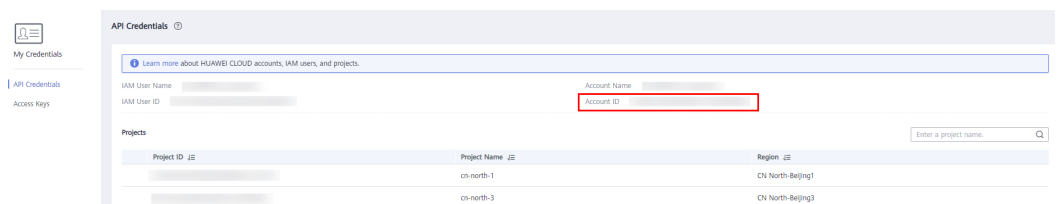
7.4 Obtaining an Account ID

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 7-3 Obtaining an account ID



7.5 How to Obtain Parameters in the API URI

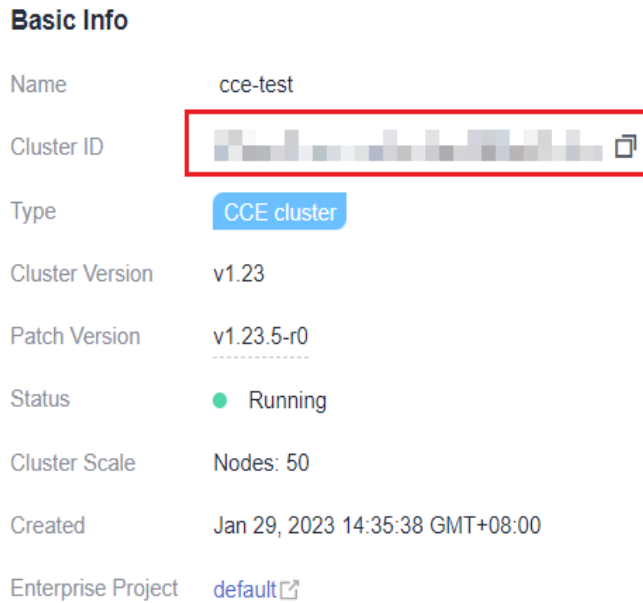
Obtaining a Project ID (project_id)

project_id indicates the project ID, which can be obtained from the console or APIs. For details, see [Obtaining a Project ID](#).

Obtaining a Cluster 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 7-4 Obtaining the cluster ID

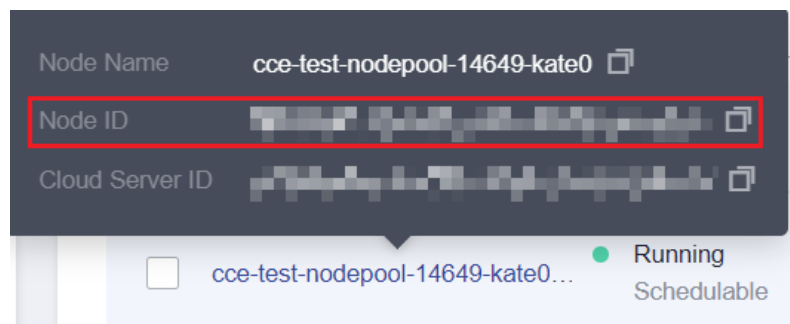


----End

Obtaining a Node ID (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 7-5 Obtaining the node ID

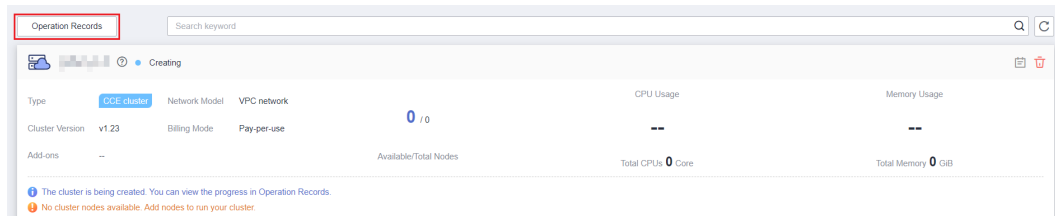


----End

Obtaining a Job ID (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.

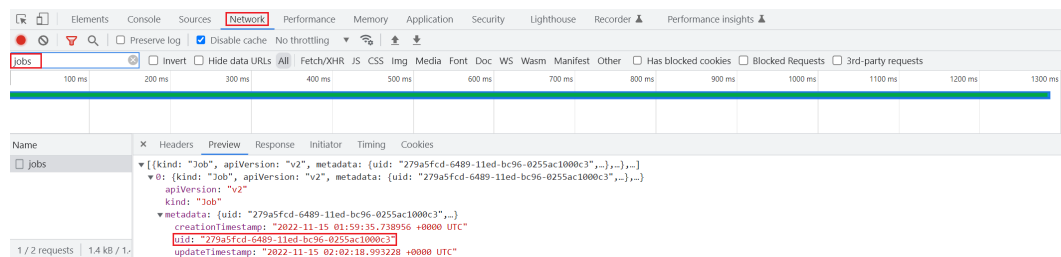
Figure 7-6 Creating a cluster



- 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 7-7 Obtaining the job ID



----End

7.6 Creating a VPC and Subnet

Context

To provide a secure and isolated network environment for CCE, create a VPC before creating a cluster.

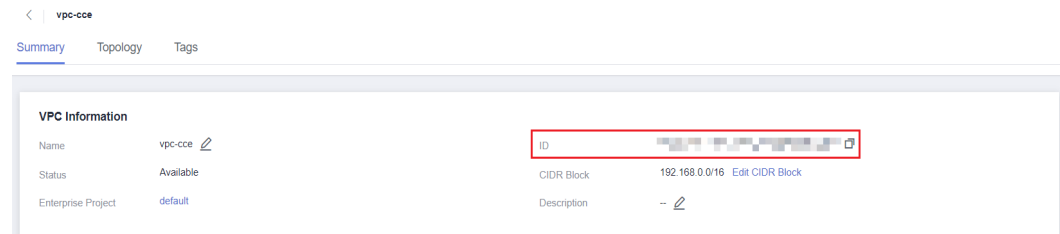
If you have already created a VPC, you do not need to create it again.

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 7-8 Obtaining the VPC ID

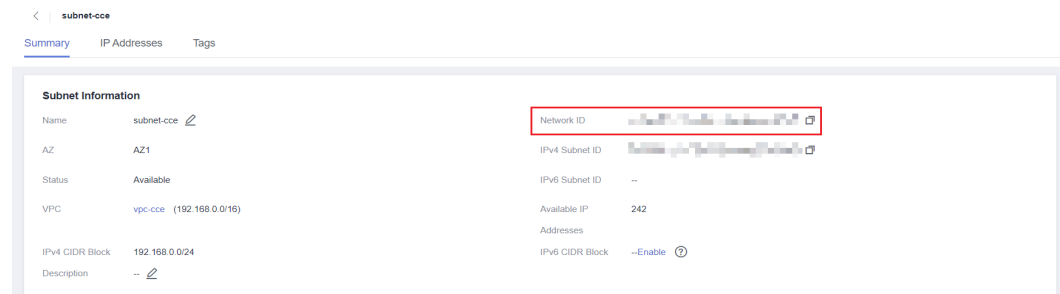


----End

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 7-9 Obtaining the network ID of a subnet



----End

7.7 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