Cloud Container Instance (CCI)

Service Overview

Issue	01
Date	2025-06-23

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions

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

Notice

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

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

Contents

1 What Is CCI?

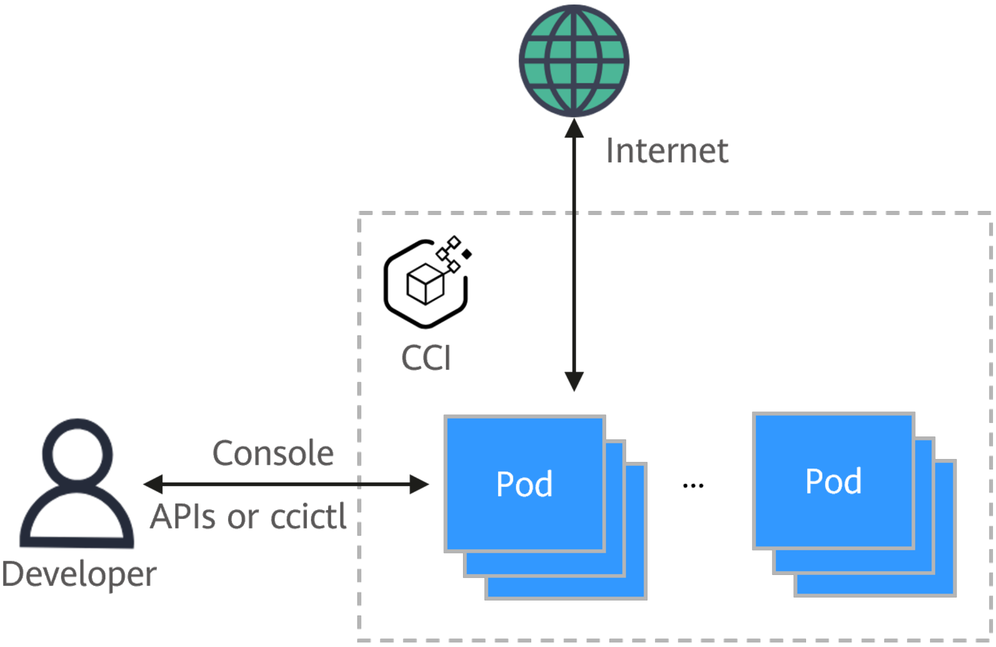
Overview

Cloud Container Instance (CCI) is a serverless container engine that allows you to run containers without creating or managing server clusters.

Traditionally, to run containerized workloads, you need to create a server cluster first. In the serverless model, a cloud provider runs servers and dynamically allocates resources so that you can build and run applications without having to worry about server statuses. This model helps you improve development efficiency and reduce IT costs.

CCI uses the serverless model that allows you to directly deploy and run containerized workloads through the console. You only need to pay for the resources consumed by the workloads.

Figure 1-1 Using CCI



Functions

One-Stop Container Lifecycle Management

CCI allows you to run containers without creating or managing server clusters. In the serverless model, you can deploy and run workloads through the console.

Different Ways for Network Access

Network access and load balancing at Layer 4 are available to meet scenario-specific needs.

Persistent Storage Volumes

CCI allows data to be stored in cloud storage. Currently, SFS Turbo can be used for persistent storage.

Excellent Auto Scaling

CCI allows you to define scaling policies and can complete auto scaling in seconds. In addition, these policies can be combined flexibly to handle traffic bursts during peak hours.

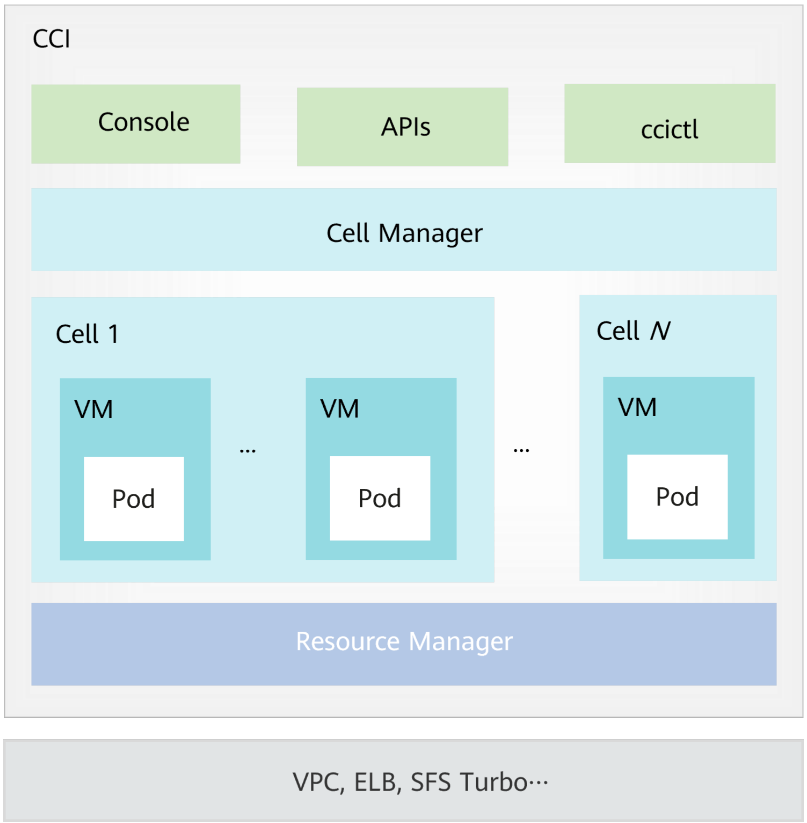
Comprehensive Container Monitoring

The resources consumed by containers are monitored, such as the CPU usage and memory usage, from which you can view the container status in real time.

Architecture

CCI is a serverless container service. CCI's converged resource pool integrates network and storage services, allowing you to easily deploy and run containerized workloads through the console or APIs.

Figure 1-2 Architecture



- CCI works with network and storage services of the cloud platform to provide robust network and storage performance.
- The QingTian architecture fueled by hardware-software synergy brings excellent performance and user experience.
- Virtualization provides security isolation and works with in-house hardware virtualization acceleration technologies and custom container OSs to allow you to create high-performance pods for running workloads.
- With unified resource management and workload scheduling, you do not need to manage clusters.
- Capabilities, such as quick workload deployment, elastic load balancing, and fast auto scaling, help iterate services quickly.

2 CCI Advantages

Out of the Box

Industry-leading serverless architecture and converged resource pool that integrates network and storage services allow you to create and run containerized workloads through the console, without creating server clusters.

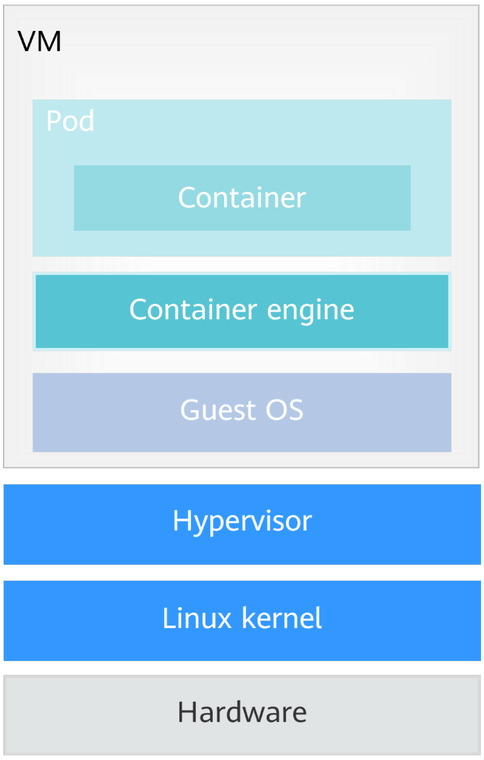
Per-Second Billing

Resources can be billed by the second to reduce costs.

High Security

CCI provides VM-level isolation without compromising the startup speed, offering you better container experience.

Figure 2-1 VM-level security isolation



3 Application Scenarios

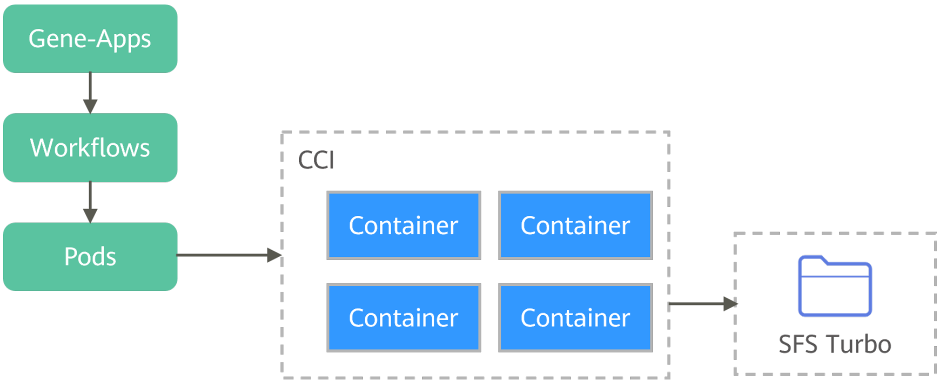
Scientific Computing

Scientific R&D in fields such as genomics and drug development requires high-performance and high-density computing. In addition, scientific computing is generally task-based and requires quick resource allocation and release. Therefore, a low-cost computing platform with automated O&M is required.

The following features make CCI suitable for computing in this scenario:

- High-performance computing and network, and high I/O storage
- Resource scaling in seconds, which minimizes resource consumption
- No O&M required for clusters and servers, greatly reducing O&M costs
- On-demand usage and billing

Figure 3-1 Scientific computing



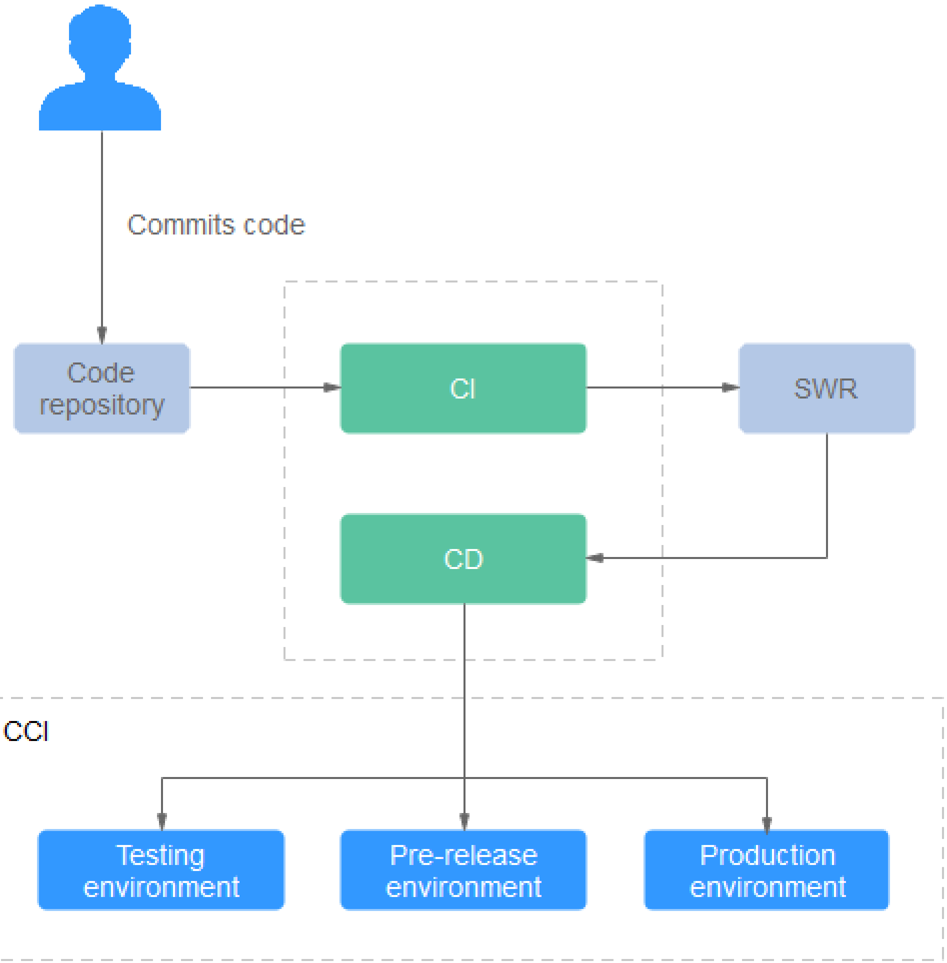
DevOps

Software development enterprises need complete DevOps processes from code submission to application deployment to improve the development efficiency. DevOps processes such as continuous integration/continuous delivery (CI/CD) are generally task-based computing and require quick resource allocation and release.

The following features make CCI suitable for computing in this scenario:

- Automation for the entire CI/CD process, with no cluster creation and maintenance required
- Image-based delivery, allowing for consistency between the development and production environments

Figure 3-2 DevOps



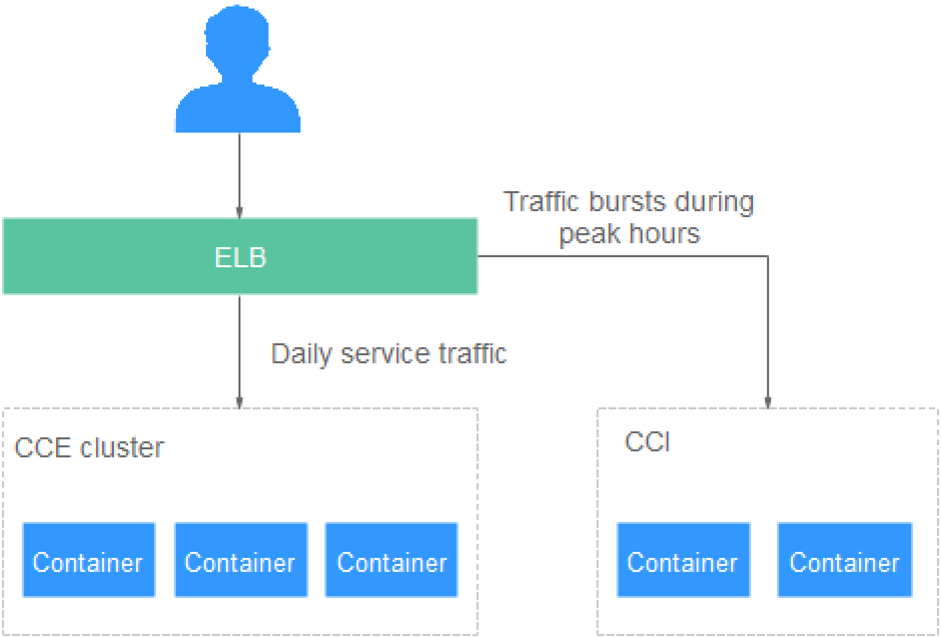
Services with Fluctuating Traffic

Some types of applications, such as applications for live video, media information, e-commerce, and online education, have obvious service peaks and troughs. For these applications, resources need to be expanded rapidly during peak hours without breaking the bank.

The following features make CCI suitable for these applications:

- **Fast auto scaling**: CCI can quickly take over services from CCE to ensure uptime during peak hours.

Figure 3-3 Auto scaling



4 Basic Concepts

CCI provides Kubernetes-like APIs based on the Kubernetes ecosystem and allows you to create associated resources through the console or APIs. Before using CCI, you are advised to understand related basic concepts to better understand CCI.

Image

A container image is a special file system that provides the programs, libraries, resources, and configuration files required for running a container. It also contains configuration parameters, for example, anonymous volumes, environment variables, and users. An image does not contain any dynamic data, and its content will not be changed after creation.

Container

The relationship between an image and a container is similar to that between a class and an instance in object-oriented programming. Images are static, while containers are running entities of images. A container can be created, started, stopped, deleted, and suspended.

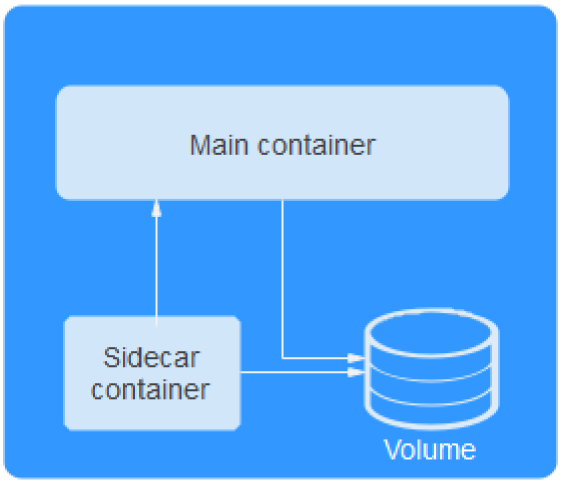
Namespace

A namespace provides a method of allocating resources among multiple users. When you have a large number of projects and personnel, you can define namespaces based on project attributes, such as production, test, and development.

Pod

A pod is the smallest deployable unit of computing that you can create and manage. A pod is a group of one or more containers, with shared storage resources, a unique IP address, and a specification for how to run the containers.

Figure 4-1 Pod



Pods can be used in either of the following ways:

- One pod that runs a single container: This is the most common use case. You can think of a pod as a wrapper around a single container, and the pod can be managed directly rather than the container.
- One pod runs multiple containers that need to work together:

Generally, pods are not created directly but are created through workloads, for example, Deployments. A workload can have multiple pods and enjoy replica management, rolling upgrade, and self-healing. Generally, pods are created from a pod template.

Init Container

An init container is a type of container that starts and exits before the application containers start in a pod. If there are multiple init containers, they will be started in the defined sequence. The data generated in the init containers can be used by the application containers because storage volumes in a pod are shared.

Init containers are designed to perform initialization tasks and can be used in multiple resources, such as Deployments and jobs.

For details, see **Init Containers**.

Label

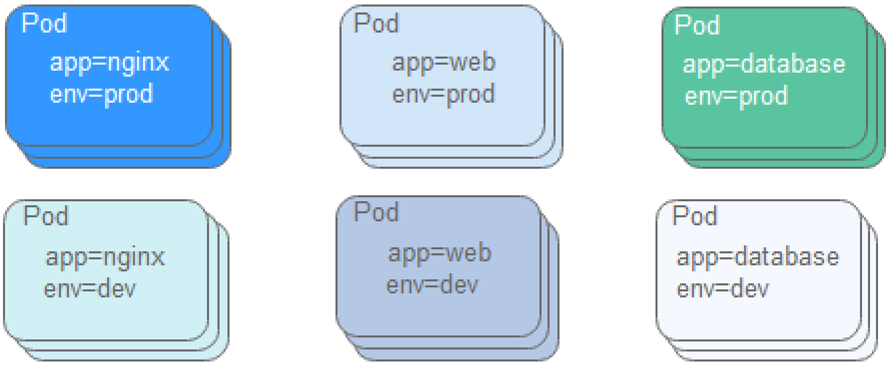
A label is a key-value pair attached to an object and is used to transfer user-defined attributes.

Labels are often used to select objects that meet conditions from a group of objects. Labels are currently the most important node grouping method.

For example, you may create labels (**tier=frontend**, **app=myapp**) to mark frontend pods and labels (**tier=backend**, **app=myapp**) to mark backend pods. You can then use selectors to select pods with specific labels and apply services or Deployments to these pods.

For details, see **Labels**.

Figure 4-2 Pods organized with labels



Deployment

Deployments are a type of workloads.

A Deployment can contain one or more pods. Each pod has the same role, and the system automatically distributes requests to the pods of a Deployment. All pods created for a Deployment share storage volumes.

When using a Deployment, you only need to describe your desired pod status. The Deployment will help you change the pod status to the target status.

For details, see **Deployments**.

Service

Pods are mortal. They can be created and destroyed. Once destroyed, they cannot be resurrected. Pod controllers create and destroy pods dynamically (for example, during scaling or rolling upgrades). Each pod obtains its own IP address, but the IP address is not always stable or dependable. This leads to a problem: if a group of pods (backends) provides services to another group of pods (frontends) inside a cluster, how do those frontends find out and connect to the corresponding backends?

A Service is a method of exposing a network application running in a pod or a group of pods as a network service. Each Service object defines a logical set of endpoints (usually pods) and policies for accessing these pods.

Consider an image processing backend that is running with three pod replicas as an example. These replicas are interchangeable. This means the frontend does not need to know which replica it calls. The pods in the backend may change, and the frontend does not need to be aware of that or keep track of the backend status. A Service enables this decoupling.

For details, see **Service**.

ConfigMap

A ConfigMap is used to store configuration data as key-value pairs or configuration files. ConfigMaps are similar to secrets, but provide a means of working with strings that do not contain sensitive information.

For details, see **ConfigMaps**.

Secret

A secret is an object for storing sensitive data such as authentication information, certificates, and private keys. A secret can be loaded to a container as environment variables when the container is started.

For details, see **Secrets**.

5 Permissions Management

If you need to assign different permissions to employees in your enterprise to access your CCI resources, Identity and Access Management (IAM) is a good choice for fine-grained permissions management. IAM provides identity authentication, permissions management, and access control, enabling secure access to your cloud resources.

With IAM, you can use your account to create IAM users, and assign permissions to the users to control their access to specific resources. For example, some software developers in your enterprise need to use CCI resources but should not be allowed to delete the resources or perform any other high-risk operations. In this scenario, you can create IAM users for the software developers and grant them only the permissions required for using CCI resources.

If your account does not require individual IAM users for permissions management, skip this topic.

CCI Permissions

By default, new IAM users do not have permissions assigned. You need to add a user to one or more groups, and attach permissions policies or roles to these groups. Users inherit permissions from the groups to which they are added and can perform specified operations on cloud services based on the permissions.

CCI is a project-level service deployed and accessed in specific regions. To assign CCI permissions to a user group, specify the scope as region-specific projects and select projects for the permissions to take effect. If **All projects** is selected, the permissions will take effect for the user group in all region-specific projects. When accessing CCI, the users need to switch to a region where they have been authorized to use this service.

You can grant users permissions by using roles and policies.

- Roles: A type of coarse-grained authorization mechanism that defines permissions related to user responsibilities. This mechanism provides only a limited number of service-level roles for authorization. Cloud services depend on each other. When using roles to assign permissions, you also need to assign dependent roles. However, roles are not an ideal choice for fine-grained authorization and secure access control.
- Policies: A type of fine-grained authorization mechanism that defines permissions required to perform operations on specific cloud resources under

certain conditions. This mechanism allows for more flexible policy-based authorization, meeting requirements for least privilege access. For example, you can grant CCI users only the permissions for managing a certain type of CCI resources. Most policies define permissions based on APIs.

Table 5-1 lists all the system-defined roles and policies supported by CCI. System-defined policies in role/policy-based authorization are not interoperable with those in identity policy-based authorization.

Table 5-1 System-defined roles and policies supported by CCI

Role/Policy Name	Description	Type
CCI FullAccess	Full permissions for CCI. Users granted these permissions can create, delete, query, and update all CCI resources.	System-defined policy
CCI ReadOnlyAccess	Read-only permissions for CCI. Users granted these permissions can only view CCI resources.	System-defined policy
CCI CommonOperations	Common user permissions for CCI. Users granted these permissions can perform all operations except creating, deleting, and modifying networks, and resources in namespaces.	System-defined policy
CCI Administrator	Administrator permissions for CCI. Users granted these permissions can create, delete, query, and update all CCI resources.	System-defined role

Table 5-2 lists the permissions granted by a CCI FullAccess policy.

Table 5-2 Permissions granted by a CCI FullAccess policy

Action	Description
cci:*:*	Perform all operations on Cloud Container Instance (CCI).
vpc:*:*	Perform all operations on Virtual Private Cloud (VPC).
elb:*:*	Perform all operations on Elastic Load Balance (ELB).
sfs:*:*	Perform all operations on Scalable File Service (SFS).
obs:*:*	Perform all operations on Object Storage Service (OBS).
evs:*:*	Perform all operations on Elastic Volume Service (EVS).
apm:*:*	Perform all operations on Application Performance Management (APM).

Action	Description
swr:*:*	Perform all operations on SoftWare Repository for Container (SWR).
nat:*:*	Perform all operations on NAT Gateway.
kms:cmk:*	Perform all operations on .

Table 5-3 lists the permissions granted by a CCI ReadOnlyAccess policy.

Table 5-3 Permissions granted by a CCI ReadOnlyAccess policy

Action	Description
cci:*:get	View details about all CCI resources.
cci:*:list	List all CCI resources.
vpc:*:get	View details about all VPC resources.
vpc:*:list	List all VPC resources.
ecs:*:get	View details about all ECS resources.
ecs:*:list	List all ECS resources.
elb:*:get	View details about all ELB resources.
elb:*:list	List all ELB resources.
sfs:*:get*	View details about all SFS resources.
sfs:*:list	List all SFS resources.
obs:*:get*	View details about all OBS resources.
obs:*:list	List all OBS resources.
evs:*:get*	View details about all EVS resources.
evs:*:list	List all EVS resources.
amp:*:get	View details about all APM resources.
apm:*:list	List all APM resources.
swr:*:get	View details about all SWR resources.
swr:*:list	List all SWR resources.
nat:*:get	View details about all NAT Gateway resources.
nat:*:list	List all NAT Gateway resources.
kms:cmk:get	Query key information.
kms:cmk:list	List all keys.

Table 5-4 lists the permissions granted by a CCI CommonOperations policy.

Table 5-4 Permissions granted by a CCI CommonOperations policy

Action	Description
cci:namespace:get	Query namespace details.
cci:namespace:list	List all namespaces.
cci:network:get	Query network details.
cci:network:list	List all networks.
cci:namespaceSubResource:*	Perform all operations on resources in namespaces.
cci:addonTemplate:*	Perform all operations on add-on templates.
cci:addonInstance:*	Perform all operations on add-on pods.
vpc:*:*	Perform all operations on VPC.
elb:*:*	Perform all operations on ELB.
sfs:*:*	Perform all operations on SFS.
obs:*:*	Perform all operations on OBS.
evs:*:*	Perform all operations on EVS.
apm:*:*	Perform all operations on APM.
swr:*:*	Perform all operations on SWR.
nat:*:*	Perform all operations on NAT Gateway.
kms:cmk:*	Perform all operations on .

Table 5-5 lists the common operations supported by system-defined permissions for CCI.

Table 5-5 Common operations supported by system-defined permissions

Operation	CCIFullAccess	CCIReadOnlyAccess	CCI CommonOperations
Creating a namespace	√	x	x
Deleting a namespace	√	x	x

Operation	CCIFullAccess	CCIReadOnlyAccess	CCI CommonOperations
Listing the namespaces	√	√	√
Querying namespace details	√	√	√
Creating a network	√	x	x
Deleting networks	√	x	x
Listing the networks	√	√	√
Querying network details	√	√	√
Updating a network	√	x	x
Creating a pod	√	x	√
Deleting pods	√	x	√
Listing the pods	√	√	√
Querying pod details	√	√	√
Updating a pod	√	x	√
Running commands in a pod	√	x	√
Querying pod logs	√	√	√
Creating a ConfigMap	√	x	√
Deleting a ConfigMap	√	x	√
Listing the ConfigMaps	√	√	√
Querying ConfigMap details	√	√	√
Updating a ConfigMap	√	x	√
Creating a secret	√	x	√
Deleting a secret	√	x	√
Listing the secrets	√	√	√
Querying secret details	√	√	√
Updating a secret	√	x	√

Operation	CCIFullAccess	CCIReadOnlyAccess	CCI CommonOperations
Creating a Service	√	x	√
Deleting a Service	√	x	√
Listing the Services	√	√	√
Querying Service details	√	√	√
Updating a Service	√	x	√
Creating a Deployment	√	x	√
Deleting a Deployment	√	x	√
Listing the Deployments	√	√	√
Querying Deployment details	√	√	√
Updating a Deployment	√	x	√
Creating an HPA policy	√	x	√
Deleting an HPA policy	√	x	√
Listing the HPA policies	√	√	√
Querying HPA policy details	√	√	√
Updating an HPA policy	√	x	√
Creating a PV	√	x	√
Deleting a PV	√	x	√
Listing the PVs	√	√	√
Querying PV details	√	√	√
Updating a PV	√	x	√
Creating a PVC	√	x	√
Deleting a PVC	√	x	√

Operation	CCIFullAccess	CCIReadOnlyAccess	CCI CommonOperations
Listing the PVCs	√	√	√
Querying PVC details	√	√	√
Updating a PVC	√	x	√
List storage classes	√	√	√
Creating an image snapshot	√	x	√
Deleting an image snapshot	√	x	√
Listing the image snapshots	√	√	√
Querying image snapshot details	√	√	√

CCI supports authorization with identity policies. **Table 5-6** lists all system-defined identity policies for CCI with identity policy-based authorization. System-defined policies in identity policy-based authorization are not interoperable with those in role/policy-based authorization.

Table 5-6 System-defined identity policies for CCI

Identity Policy Name	Description	Type
CCIFullAccessPolicy	Full permissions for CCI	System-defined identity policy
CCIReadOnlyPolicy	Read-only permissions for CCI	System-defined identity policy

The following table lists the permissions of the CCIFullAccessPolicy identity policy.

Table 5-7 Permissions assigned by CCIFullAccessPolicy

Action	Description
cci:*:*	Perform all operations on CCI.
vpc:subnets:create	Create a VPC subnet.
vpc:subnets:get	Query VPC subnet details.

Action	Description
vpc:subnets:update	Update a VPC subnet.
vpc:subnets:delete	Delete a VPC subnet.
vpc:vpcs:create	Create a VPC.
vpc:vpcs:get	Query VPC details.
vpc:vpcs:list	List the VPCs.
vpc:vpcs:update	Update a VPC.
vpc:vpcs:delete	Delete a VPC.
vpc:ports:get	Query VPC port details.
vpc:ports:list	List VPC ports.
vpc:quotas:list	Query VPC resource quotas.
vpc:securityGroups:get	Query security group details.
vpc:securityGroupRules:get	Query security group rule details.
swr:namespace:list*	(SWR Shared Edition) List the organizations.
swr:namespace:get*	(SWR Shared Edition) Query organization permissions and details.
swr:repo:list*	(SWR Shared Edition) List all image resources.
swr:repo:get*	(SWR Shared Edition) Query details about all image resources.
swr:repo:download	(SWR Shared Edition) Download images.
swr::listQuotas	(SWR Shared Edition) Query quotas.
swr::getDomainOverview	(SWR Shared Edition) Query the tenant resource overview.
swr::getDomainResourceReports	(SWR Shared Edition) Query tenant resource statistics.
swr:instance:get*	Query details about all SWR instance resources.
swr:instance:list*	List all SWR instance resources.

Table 5-8 Permissions assigned by CCIReadOnlyPolicy

Action	Description
cci:*:get*	Query details about all CCI resources.

Action	Description
cci:*:list*	List all CCI resources.
vpc:subnets:get	Query VPC subnet details.
vpc:vpcs:get	Query VPC details.
vpc:vpcs:list	List the VPCs.
vpc:ports:get	Query VPC port details.
vpc:ports:list	List VPC ports.
vpc:quotas:list	Query VPC resource quotas.
vpc:securityGroups:get	Query security group details.
vpc:securityGroupRules:get	Query security group rule details.
swr:namespace:list*	(SWR Shared Edition) List the organizations.
swr:namespace:get*	(SWR Shared Edition) Query organization permissions and details.
swr:repo:list*	(SWR Shared Edition) List all image resources.
swr:repo:get*	(SWR Shared Edition) Query details about all image resources.
swr:repo:download	(SWR Shared Edition) Download images.
swr::listQuotas	(SWR Shared Edition) Query quotas.
swr::getDomainOverview	(SWR Shared Edition) Query the tenant resource overview.
swr::getDomainResourceReports	(SWR Shared Edition) Query tenant resource statistics.
swr:instance:get*	Query details about all SWR instance resources.
swr:instance:list*	List all SWR instance resources.

Table 5-9 lists the common operations supported by system-defined identity policies for CCI.

Table 5-9 Common operations supported by system-defined identity policies

Operation	CCIFullAccessPolicy	CCIReadOnlyPolicy
Creating a namespace	√	x

Operation	CCIFullAccessPolicy	CCIReadOnlyPolicy
Deleting a namespace	√	x
Listing the namespaces	√	√
Querying namespace details	√	√
Creating a network	√	x
Deleting a network	√	x
Listing the networks	√	√
Querying network details	√	√
Updating a network	√	x
Creating a pod	√	x
Deleting a pod	√	x
Listing the pods	√	√
Querying pod details	√	√
Updating a pod	√	x
Running commands in a pod	√	x
Querying pod logs	√	x
Creating a ConfigMap	√	x
Deleting a ConfigMap	√	x
Listing the ConfigMaps	√	√
Querying ConfigMap details	√	√
Updating a ConfigMap	√	x
Creating a secret	√	x
Deleting a secret	√	x
Listing the secrets	√	√
Querying secret details	√	√
Updating a secret	√	x
Creating a Service	√	√
Deleting a Service	√	√
Listing the Services	√	√
Querying Service details	√	√

Operation	CCIFullAccessPolicy	CCIReadOnlyPolicy
Updating a Service	√	x
Creating a Deployment	√	x
Deleting a Deployment	√	x
Listing the Deployments	√	√
Querying Deployment details	√	√
Updating a Deployment	√	x
Creating an HPA policy	√	x
Deleting an HPA policy	√	x
Listing the HPA policies	√	√
Querying HPA policy details	√	√
Updating an HPA policy	√	x
Creating a PV	√	x
Deleting a PV	√	x
Listing the PVs	√	√
Querying PV details	√	√
Updating a PV	√	x
Creating a PVC	√	x
Deleting a PVC	√	x
Listing the PVCs	√	√
Querying PVC details	√	√
Updating a PVC	√	x
Listing the storage classes	√	√
Creating an image snapshot	√	x
Deleting an image snapshot	√	x
Listing the image snapshots	√	√
Querying image snapshot details	√	√

The following table lists the actions associated with CCI fine-grained policies.

Table 5-10 Actions associated with CCI fine-grained policies

Action	Description
cci:namespace:create	Create a namespace.
cci:namespace:delete	Delete a namespace.
cci:namespace:list	List the namespaces.
cci:namespace:get	Query namespace details.
cci:network:create	Create a network.
cci:network:delete	Delete a network.
cci:network:list	List the networks.
cci:network:get	Query network details.
cci:network:update	Update a network.
cci:pod:create	Create a pod.
cci:pod:delete	Delete a pod.
cci:pod:list	List the pods.
cci:pod:get	Query pod details.
cci:pod:update	Update a pod.
cci:pod:exec	Run commands in a pod.
cci:pod:getLog	Query pod logs.
cci:configmap:create	Create a ConfigMap.
cci:configmap:delete	Delete a ConfigMap.
cci:configmap:list	List the ConfigMaps.
cci:configmap:get	Query ConfigMap details.
cci:configmap:update	Update a ConfigMap.
cci:secret:create	Create a secret.
cci:secret:delete	Delete a secret.
cci:secret:list	List the secrets.
cci:secret:get	Query secret details.
cci:secret:update	Update a secret.
cci:service:create	Create a Service.
cci:service:delete	Delete a Service.
cci:service:list	List the Services.

Action	Description
cci:service:get	Query Service details.
cci:service:update	Update a Service.
cci:deployment:create	Create a Deployment.
cci:deployment:delete	Delete a Deployment.
cci:deployment:list	List the Deployments.
cci:deployment:get	Query Deployment details.
cci:deployment:update	Update a Deployment.
cci:horizontalpodautoscaler:create	Create an HPA policy.
cci:horizontalpodautoscaler:delete	Delete an HPA policy.
cci:horizontalpodautoscaler:list	List the HPA policies.
cci:horizontalpodautoscaler:get	Query HPA policy details.
cci:horizontalpodautoscaler:update	Update an HPA policy.
cci:persistentvolume:create	Create a PV.
cci:persistentvolume:delete	Delete a PV.
cci:persistentvolume:list	List the PVs.
cci:persistentvolume:get	Query PV details.
cci:persistentvolume:update	Update a PV.
cci:persistentvolumeclaim:create	Create a PVC.
cci:persistentvolumeclaim:delete	Delete a PVC.
cci:persistentvolumeclaim:list	List the PVCs.

Action	Description
cci:persistentvolumeclaim:get	Query PVC details.
cci:persistentvolumeclaim:update	Update a PVC.
cci:storageclass:list	List the storage classes.
cci:imagesnapshot:create	Create an image snapshot.
cci:imagesnapshot:delete	Delete an image snapshot.
cci:imagesnapshot:list	List the image snapshots.
cci:imagesnapshot:get	Query image snapshot details.

6 Notes and Constraints

This topic describes the constraints on using CCI.

Image Pull Constraints

VPC endpoints are required to pull images from SWR. Currently, VPC endpoints cannot be created automatically. Therefore, before using CCI, you need to manually create VPC endpoints. If there are no VPC endpoints, an error message will be displayed indicating that the image fails to be pulled when you create a pod or Deployment.

You need the following VPC endpoints:

- To pull images from a repository of SWR Enterprise Edition, you need to purchase a VPC endpoint for OBS.
- To pull images from the SWR public image repository, you need to purchase a VPC endpoint for SWR and a VPC endpoint for OBS.

  For details, see **Purchasing VPC Endpoints**.

Constraints on Pod Specifications

The following table lists the pod specifications.

Table 6-1 Supported pod specifications

Container vCPUs	Container Memory (GiB)
0.25	0.5, 1, and 2
0.5	0.5, 1, 2, 3, and 4
1	1 to 8 (increment: 1 GiB)
2	2 to 16 (increment: 1 GiB)
4	4 to 32 (increment: 1 GiB)
8	8 to 64 (increment: 4 GiB)
16	16 to 128 (increment: 8 GiB)

Container vCPUs	Container Memory (GiB)
32	32, 64, 128, and 256
48	96, 192, and 384
64	128, 256, and 512

NOTE

For pods running on CCI, the OS and CCI occupy some underlying resources. In some extreme scenarios, the actual memory usage of a pod may not reach the memory in the pod specifications. If you need to use all resources defined in the specifications, CCI allows you to reserve system overhead. For details, see **Increasing Reserved System Overhead**.

Constraints on Pod Storage Space

If no EVS disk is mounted, application data is stored in container's rootfs. The following table lists the default storage space of each pod.

Table 6-2 Default storage space of each pod

Pod Type	Storage Space
General	30 GiB

NOTE

As the OS and CCI occupy some underlying resources, the disk usage cannot reach the default storage space in actual use. In addition, the larger the pod specifications you select, the more the system resources are occupied. By default, if the ephemeral storage of a pod exceeds 20 GiB, you can expand the pod storage as needed. For details, see **Adding Ephemeral Storage Capacity**.

Constraints

- For Services of the LoadBalancer type, only dedicated load balancers are available. If a Service of the LoadBalancer type is used, the pods can have IPv4 IP addresses.
- If the CCE Cloud Bursting Engine for CCI add-on is used to schedule the workloads to CCI, dedicated load balancers can be configured for ingresses and Services of the LoadBalancer type. The CCE Cloud Bursting Engine for CCI add-on does not support Services of the LoadBalancer Services if its version is earlier than 1.5.5.

7 Region and AZ

Concept

A region and availability zone (AZ) identify the location of a data center. You can create resources in a specific region and AZ.

- 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.
- An AZ contains one or more physical data centers. Each AZ has independent cooling, fire extinguishing, moisture-proof, and electricity facilities. Within an AZ, computing, network, storage, and other resources are logically divided into multiple clusters.

Selecting a Region

- Relationship between cloud services

  When using multiple cloud services, note the following restrictions:

  – ECSs, RDS instances, and OBS buckets in different regions cannot communicate with each other through an internal network.

  – ECSs in different regions cannot be bound to the same load balancer.

If your target users are in Europe, select the **EU-Dublin** region.

Selecting an AZ

When deploying resources, consider your applications' requirements on disaster recovery (DR) and network latency.

- For high DR capability, deploy resources in different AZs within the same region.
- For lower network latency, deploy resources in the same AZ.

8 Related Services

CCI needs to be interconnected with other cloud services and requires permissions to access these cloud services.

- **SoftWare Repository for Container (SWR)**

  SWR provides easy, secure, and reliable management of container images throughout their lifecycles, facilitating quick deployment of containerized services.

  You can create workloads using SWR images.

- **Virtual Private Cloud (VPC)**

  VPC allows you to isolate online resources with virtual private networks. You can configure a CIDR block for a VPC, create subnets in a VPC, and configure security groups, assign EIPs, and allocate bandwidth for instances in a VPC.

  When creating a namespace, you must associate it with a VPC. All containers to be created in the namespace will run in this VPC.

- **Elastic Load Balance (ELB)**

  ELB automatically distributes incoming traffic to multiple backend servers to balance the loads. It enhances an application's fault tolerance and capabilities of providing services externally.

  An elastic load balancer allows access to containers from an external network.

- **Scalable File Service Turbo (SFS Turbo)**

  SFS Turbo provides high-performance file storage (NAS) that can be scalable to 320 TB. It can provide high availability and durability for workloads dealing with massive small files and applications that require low latency and high IOPS.

  You can mount file systems to containers for persistent storage when creating workloads.

- **Log Tank Service (LTS)**

  LTS is a log platform that features high performance, cost effectiveness, rich functions, and high availability. It provides multiple modes to ingest massive logs to LTS. It is integrated with log search, SQL analysis, and log processing engines.

  LTS helps collect container logs and dumps log files so that you can view and search for logs later.

- **Cloud Trace Service (CTS)**

  CTS records operations on your cloud resources, allowing you to query, audit, and backtrack resource operation requests initiated from the management console or open APIs as well as responses to those requests.