Cloud Container Instance (CCI)

FAQs

Issue	01
Date	2025-11-11

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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

Notice

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

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

Huawei Cloud Computing Technologies Co., Ltd.

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

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

Contents

1 Basic Concepts

What Is CCI?

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

With the serverless architecture, you can focus on building and operating applications without having to create or manage servers, or worrying about server health. All you have to do is to specify resource requirements (such as the required vCPUs and memory). This gives you a more focused approach to business needs and helps you reduce management and maintenance costs. Traditionally, to run containerized workloads using Kubernetes, you need to create a Kubernetes cluster first. With CCI, you can create and run containerized workloads using the console without creating or managing Kubernetes clusters. You only pay for the resources used by the containers.

CCI provides the following functions:

- Automated continuous delivery (CD)

  CCI can run the container image generated by the CI process in one click, which ensures that the CI/CD process is fully automated.
- Fully hosted workload runtime

  The runtime of Deployments is fully hosted to ensure that applications can run stably.
- Ultra-fast auto scaling

  You can create custom auto scaling policies for automatic scaling within seconds.
- High availability for applications

  Multiple pods can provide services externally at the same time, and global load balancing ensures that no pods are overloaded.
- Container status monitoring

  The health of containers can be checked, and container metrics are monitored in real time.
- Persistent data storage

  Storage volumes can be mounted to containers for persistent data storage.

What Are the Differences Between CCI and CCE?

Huawei Cloud provides enterprise-class container services with high performance, availability, and security. There are two types of container services that have been officially certified by CNCF and developed based on the Kubernetes ecosystem: Cloud Container Engine (CCE) and CCI.

Two cloud services differ from each other in the following aspects: **service introduction**, **cluster creation**, **billing modes**, and **application scenarios**.

- Service introduction

  Table 1-1 Introduction to CCE and CCI

CCE	CCI
CCE provides highly scalable, high-performance, enterprise-class Kubernetes clusters and supports Docker containers. CCE is a one-stop container platform that provides full-stack container services from Kubernetes cluster management, lifecycle management of containerized applications, application service mesh, and Helm charts to add-on management, application scheduling, and monitoring and O&M. With CCE, you can easily deploy, manage, and scale containerized applications on Huawei Cloud.	Traditionally, to run containerized workloads using Kubernetes, you need to create a Kubernetes cluster first. CCI is a serverless container service that allows you to run containers without creating or managing server clusters. With CCI, you only need to manage containerized services. You can quickly create and run workloads on CCI without managing clusters and servers. Because of the serverless architecture, CCI frees you from containerized application O&M and allows you to focus on the services.

- Cluster creation

  Table 1-2 Cluster creation

CCE	CCI
CCE is a hosted Kubernetes service for container management. It allows you to create native Kubernetes clusters with just a few clicks. You need to create clusters and nodes on the console, but you do not need to manage master nodes.	CCI provides a serverless container engine. When deploying containers on Huawei Cloud, you do not need to purchase and manage ECSs, eliminating the need for O&M and management. You can start applications without the need to create clusters, master nodes, or worker nodes.

- Billing modes

Table 1-3 Billing modes

Item	CCE	CCI
Pricing	Resources	vCPUs (vCPU-hour) and memory (GiB-hour)
Billing mode	Yearly/Monthly or pay-per-use	Pay-per-use and packages
Minimum pricing unit	Hour	Second

- Application scenarios

  Table 1-4 Application scenarios

CCE	CCI
All scenarios. Generally, CCE runs large-scale and long-term stable applications, such as e-commerce, service middle-end, and IT system.	Batch computing, high-performance computing, scale-out during traffic spikes, and CI/CD tests

- Scheduling workloads from CCE to CCI

  The CCE Cloud Bursting Engine for CCI add-on can schedule Deployments, StatefulSets, and jobs running on CCE to CCI when there are traffic spikes. This can reduce consumption caused by cluster scaling.

  The following are benefits of this add-on:

  – Automatic pod scaling within seconds: When CCE cluster resources are insufficient, there is no need to add nodes to the CCE cluster, and the CCE Cloud Bursting Engine for CCI add-on automatically creates pods in CCI, eliminating the overhead of resizing the CCE cluster.
  – CCI seamlessly works with Huawei Cloud SWR so that you can use public and private images in SWR repositories.
  – CCI supports event synchronization, monitoring, logging, remote command execution, and status query for pods.
  – You can view the capacity information about virtual elastic nodes.
  – Service networks of CCE pods and CCI pods can communicate with each other.

What Is an Environment Variable?

An environment variable is a variable whose value can affect the way a running container will behave. You can modify environment variables even after workloads are deployed, increasing flexibility in workload configuration.

The result of setting environment variables in CCI is the same as that of specifying **ENV** in a Dockerfile.

What Is Mcore?

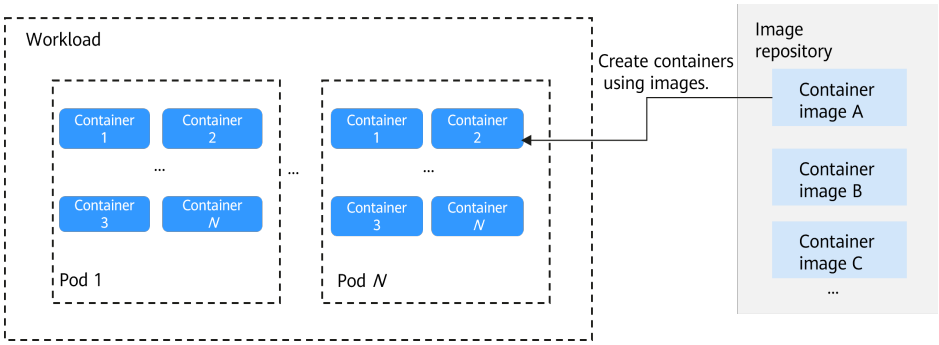
A millicore, abbreviated as mcore, is one-thousandth of a vCPU. Generally, the vCPU usage of a containerized workload is measured in mcores.

What Are the Relationships Between Images, Containers, and Workloads?

- An image is a special file that includes all the programs, libraries, resources, and configurations for running containers. It also includes some parameters required for the runtime, such as anonymous volumes, environment variables, and users. An image does not contain any dynamic data, and its content remains unchanged after being built.
- A container is a runtime instance of an image. An image is like a class and a container is like an instance in the object-oriented program design. A container can be created, started, stopped, deleted, or suspended.
- A workload is an application running on one or more pods. A pod consists of one or more containers. Each container is created from a container image.

The following figure shows the relationships between images, containers, and workloads.

Figure 1-1 Relationships between images, containers, and workloads



What Are vCPU-Hours in CCI Packages?

1 vCPU-hour = 1 × 3,600 vCPU-seconds

One vCPU-hour means that a vCPU is used for one hour.

One vCPU-second means that a vCPU is used for one second.

Case 1:

If a Deployment uses 2.5 vCPUs for two consecutive hours, 5 vCPU-hours are used: 2.5 vCPUs × 2 hours = 5 vCPUs × 3,600 seconds.

Case 2:

If you purchase a package with 730 vCPU-hours, you can allow containers to run 730 vCPUs for one hour or run one vCPU for 730 hours.

2 Workload Abnormalities

What Do I Do If There Is a Workload Abnormality?

If a workload is in the abnormal state, check the events first.

Log in to the CCI console. In the navigation pane on the left, choose **Workloads**. Click the name of the abnormal workload. On the details page that is displayed, view the latest event.

Image Pull Failure

If there is an event indicating that the image failed to be pulled, check the following items to locate the fault:

- **imagePullSecret** is not specified when you create a workload.

  For example, when creating a Deployment named **nginx**, check whether the YAML file contains the **imagePullSecrets** field, which indicates the secret name during image pull.

  To pull an image from **SWR**, set this field to **imagepull-secret**.

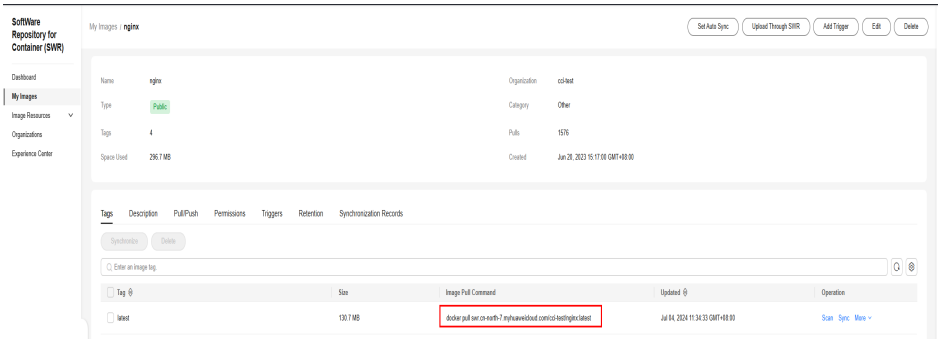
```
apiVersion: cci/v2
kind: Deployment
metadata:
  name: nginx
  namespace: nginx-ns
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx:alpine
        imagePullPolicy: Always
        name: nginx
      imagePullSecrets:
        - name: imagepull-secret
```

- The image address is incorrect.

CCI allows you to create workloads using images pulled from **SWR**.

SWR images can be obtained using the image pull command. After an image is pushed, you can obtain its address.

Figure 2-1 Image address



- IAM users do not have sufficient permissions to pull images.

  If you have enabled Enterprise Project Management Service (EPS), your account needs to assign IAM users the permissions to access SWR so that the IAM users can pull private images in your account.

  You can assign permissions to IAM users in either of the following ways:

  – Assign permissions on the details page of an image. With granted permissions, IAM users can read, edit, and manage this image. For details, see **Granting Permissions of a Specific Image**.

  – Assign permissions on the details page of an organization. With granted permissions, IAM users can read, edit, and manage all images in the organization. For details, see **Granting Permissions for an Organization**.

- The Docker version used for image packaging is outdated.

  The image failed to be pulled, and the following error information is displayed:

  ```
  failed to pull and unpack image "****": failed to unpack image on snapshotter devmapper: failed to extract layer
  sha256:xxxxxx: failed to get reader from content store: content digest sha256:xxxxxx: not found
  ```

  Cause: The Docker version used to build the image is outdated (earlier than v1.10). Some image packaging standards are no longer supported by the community.

  Solution: Use Docker v1.11 or later to rebuild the image and push it to SWR, upgrade the workload image tag and pull it again.

- VPC endpoints are not configured.

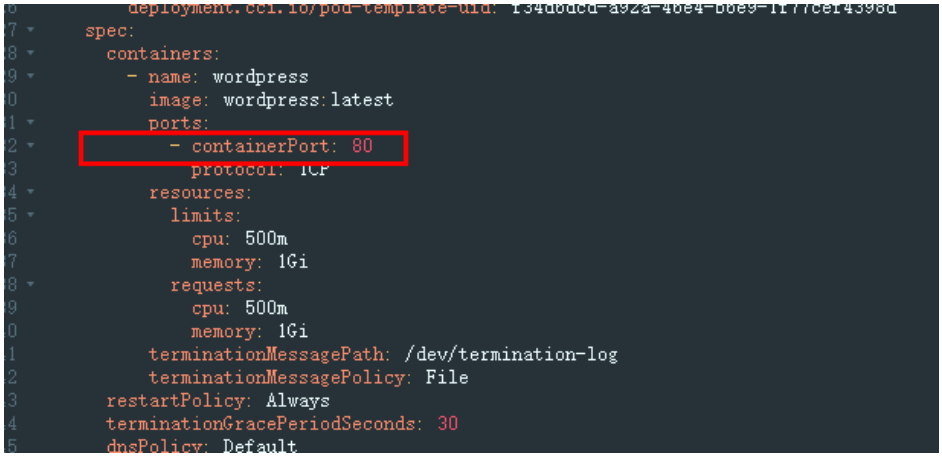
  For details, see **Purchasing VPC Endpoints**.

  Solution: If you use a repository of SWR Enterprise Edition, configure a VPC endpoint for OBS and create the workload again. If you use an SWR public image repository, configure two VPC endpoints (one for OBS and one for SWR) and create the workload again.

Container Restart Failure

If there is an event on the details page of a workload indicating that the container failed to restart, perform the following operations to locate the fault:

- Check whether there are port conflicts.
  a. Click the abnormal workload.
  b. Locate the pod where the container failed to start and click **View YAML** to check whether the container port conflicts with that of another container.



  Solution: Create the workload again and configure a correct port. Ensure that the port does not conflict with other ports.

- Check whether the workload is abnormal.

  Check whether the workload startup command is correctly executed or there is workload fault.

  Solution: Create the workload again and configure a correct startup command.

- Check whether the workload health check failed.

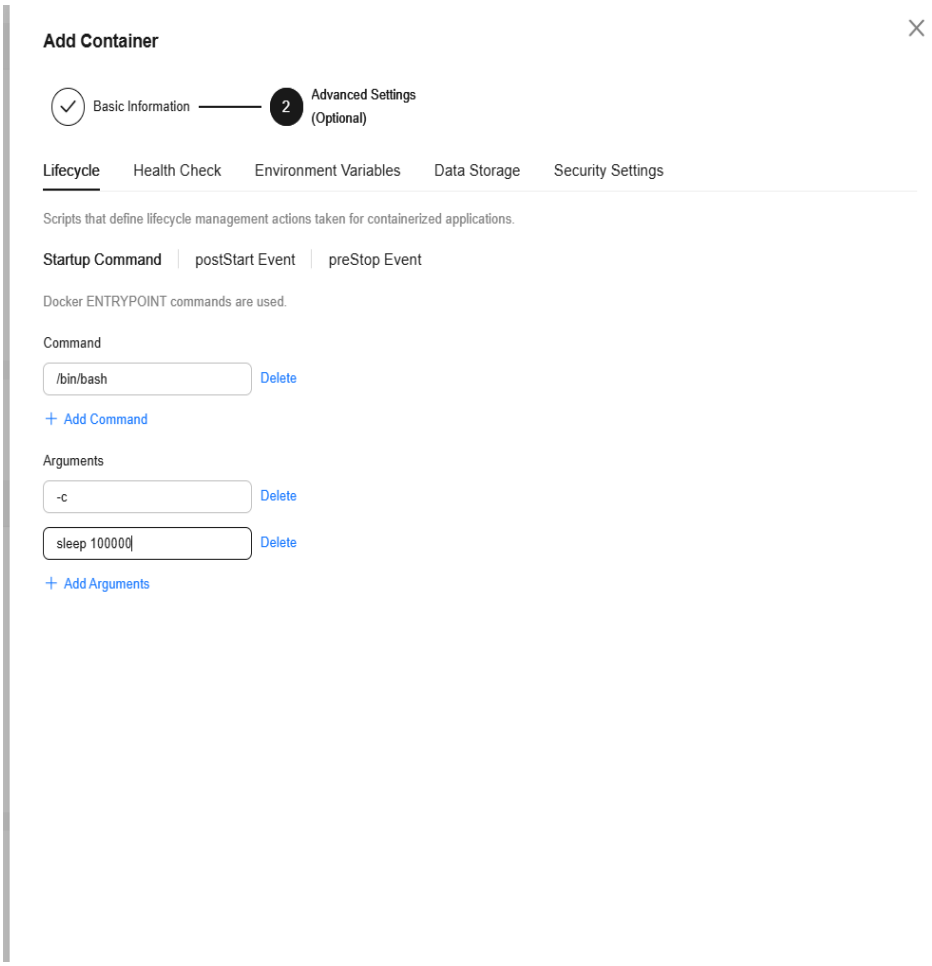
  If a liveness probe is configured for the workload and the number of health check failures exceeds the failure threshold, the container will be restarted. On the workload details page, if Kubernetes events contain "Liveness probe failed: ……", the health check fails.

  Solution: Reconfigure the health check.

Other Check Items

If the workload is abnormal, the application running in the container may fail to start. In this case, you can manually run the startup command and locate and rectify the fault based on the error message. Take the following steps:

1. Configure the startup command shown in the figure for the workload. In this way, the application will not be started after the pod is started, and no operation will be performed.



NOTE

Before running the startup command, ensure that the **/bin/bash** command in the image is available.

2. After the pod is started, locate the pod and click **View Terminal** in the **Operation** column. Then select the corresponding container, manually run the startup command, and locate and rectify the fault based on the error message.

3 Workloads

Why Can't the Service Performance Meet the Expectation?

CCI underlying resources are shared by multiple tenants. To ensure service stability, there are traffic control on the underlying resources such as disk I/O. For containers, the read and write operations on the **rootfs** directory and the standard output of workload logs are limited. If the service performance does not meet your expectations, locate the fault based on the following causes:

- **Service containers print a large number of log files to stdout streams.**

  CCI limits the forwarding traffic of standard output. If there are more 1-MB service logs per second, a log volume should be used to report logs to Application Operations Management (AOM). For details, see **Log Management**. Alternatively, you can output logs to an SFS Turbo volume and use sidecar containers to run an open-source component such as Fluent Bit to report logs to your self-managed log centers. If a large number of logs are printed to stdout streams, the service performance may be affected due to the limit.

- **Service containers have high I/O read and write operations on the rootfs disk.**

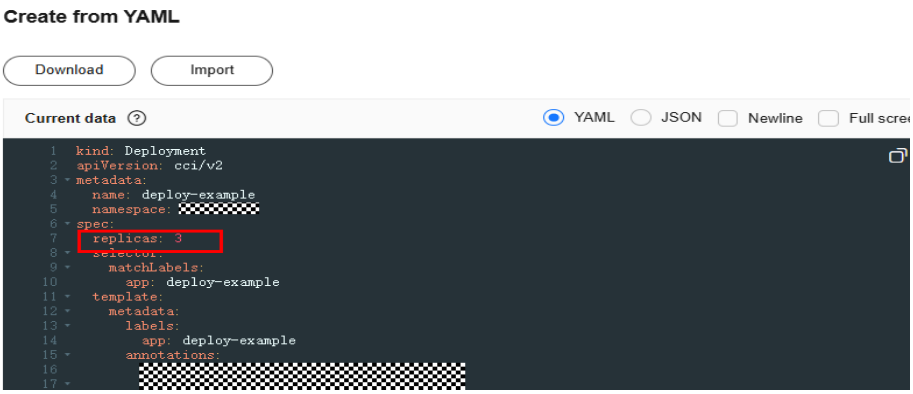
  CCI limits the I/O of the container's system disk (**rootfs**). If a process may have high disk I/O operations (bandwidth > 6 Mbit/s, IOPS > 1,000) or is sensitive to the disk I/O performance, do not execute it in **rootfs**. For example, frequently printing logs to **rootfs** may cause frequent read and write operations on it. You can place files that are not frequently read and written and service-related configuration files in **rootfs**. For high I/O file operations, select log volumes or FlexVolume (created or deleted with the pod), or persistent SFS Turbo volumes based on your service requirements. High I/O operations in **rootfs** may affect the service performance due to the disk I/O limit.

Rectify the fault in a timely manner to prevent the service performance from being affected.

How Do I Set the Quantity of Pods?

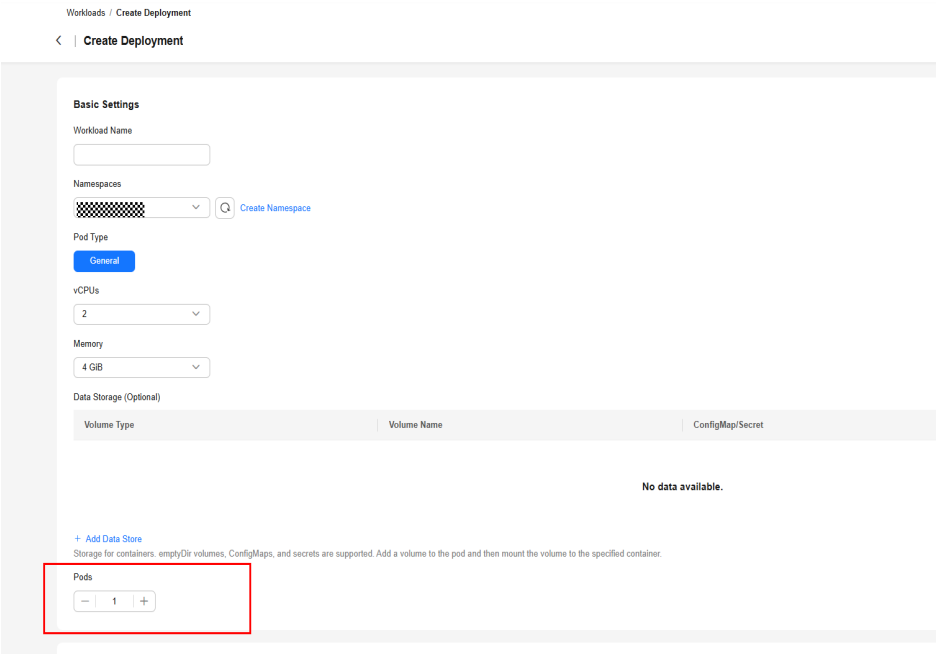
- You can set the number of pods when creating a workload.
  - When creating a workload using YAML, you can directly set the number of pods.

Figure 3-1 Setting the number of pods using YAML



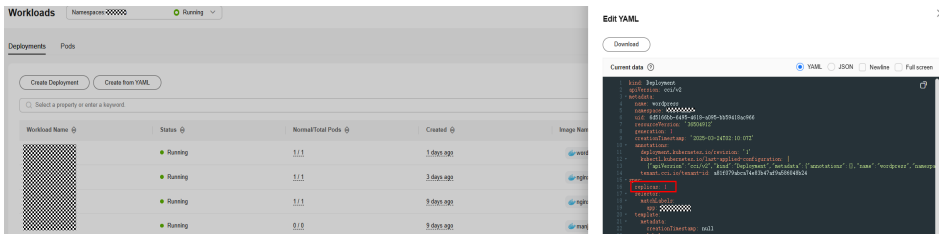
– When creating a workload on the console, you can specify the number of pods.

Figure 3-2 Setting the number of pods on the console



- After a workload is created, click **Edit YAML** and modify **replicas**.

Figure 3-3 Modifying the number of pods in the YAML file



How Do I Set Probes for a Workload?

CCI supports liveness, readiness, and startup probes. You can set them when creating a workload. For details, see **Setting Health Check Parameters**.

How Do I Configure an Auto Scaling Policy?

CCI supports custom auto scaling policies. For details, see **Scaling a Workload**.

What Do I Do If the Container Created from the Sample Image Fails to Run?

If you have enabled SWR but have not pushed any image to SWR, SWR will build an image named **sample**. However, this image cannot run. You are advised to use an image on the **Open Source Images** tab to create a workload.

How Do I View the Pods After I Delete a Deployment by Calling the API?

The value of the **propagationPolicy** field in the API request for deleting a Deployment indicates whether pods are deleted along with the Deployment. This field can be set to **Foreground** or **Background**.

Does CCI Support Container Startup in Privileged Mode?

Currently, CCI does not support the privileged mode.

In other scenarios, you should use security contexts for fine-grained permissions control, thereby ensuring the security and reliability of the container running environment.

Why Does the Anti-Affinity Policy of Pods Across Deployments Not Take Effect in CCI?

CCI does not support hard anti-affinity of pods across Deployments.

If you want to schedule pods of different Deployments to different AZs, you can use node anti-affinity.

Why Does Pod Affinity Not Take Effect in CCI?

CCI does not support pod affinity. Only pod anti-affinity is supported.

4 Image Repositories

Can I Export or Pull Public Images?

You cannot export or pull images uploaded by other users.

- To pull official images, run the corresponding container command. For example, to pull the Nginx image, run the following command:

  docker pull nginx

- To pull the images that you have pushed to SWR, take the following steps:
  a. Log in to the SWR console.
  b. In the navigation pane, choose **My Images**. Click the image to be pulled. The image details page is displayed.
  c. Click the **Pull/Push** tab, and run the **docker pull** command to pull the desired image as prompted.

How Do I Create a Container Image?

You can use Dockerfile to create a container image for a simple web workload. After you create a containerized workload using an official Nginx image, the default Nginx welcome page is displayed. The following describes how to create an image to change the default welcome message to "Hello, CCI!"

Step 1 Log in to the VM running Docker as **root**.

Step 2 Create a file named **Dockerfile**.

mkdir mynginx

cd mynginx

touch Dockerfile

Step 3 Edit the **Dockerfile** file.

vi Dockerfile

Example file content:

```
FROM nginx
RUN echo '<h1>Hello,CCI!</h1>' > /usr/share/nginx/html/index.html
```

Where,

- FROM statement: specifies that an Nginx image is used as a base image.
- RUN statement: indicates that the **echo** command is executed to display "Hello, CCI!"

Step 4 Build a container image.

docker build -t nginx:v3 .

Step 5 Check the created image. The command output shows that the image has been created with a tag of v3.

docker images

----End

How Do I Push Images to SWR?

- **Pushing an Image Through a Container Engine Client**
- **Uploading an Image Through the SWR Console**

Does CCI Provide Base Container Images?

CCI's image repository is provided by SWR, which provides base container images.

Does CCI Administrator Have the Permission to Push Image Packages?

To push images for CCI, you need to use SWR.

You also need to add the swr:repo:upload permission for the account.

What Permissions Are Required for Pushing Image Packages for CCI?

To push images for CCI, you need to use SWR.

You need the swr:repo:upload permission. For details, see **SWR permissions**.

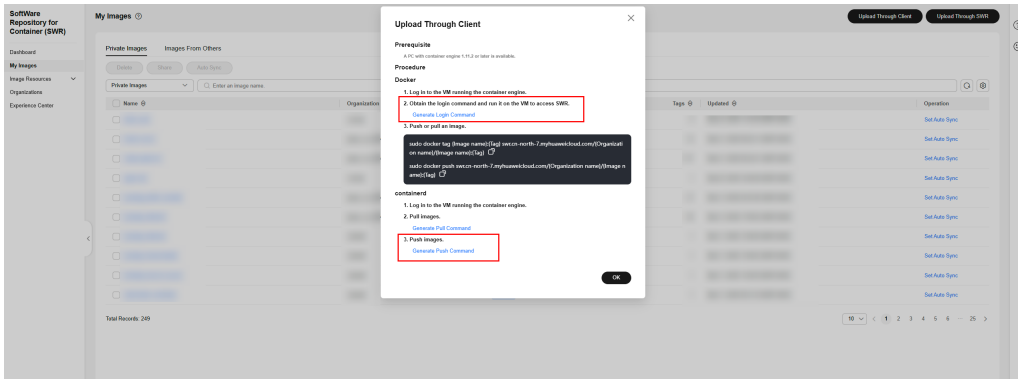
For details about how to push an image, see **Pushing an Image Through a Container Engine Client** or **Uploading an Image Through SWR Console**.

What Do I Do If Authentication Is Required During Image Push?

To push images for CCI, you need to use SWR.

To push images to SWR, you need the permission to access SWR. For details about how to push images, see **Pushing an Image Through a Container Engine Client**.

Figure 4-1 Pushing images



Is There a Way for CCI to Avoid Image Pull During Workload Startup?

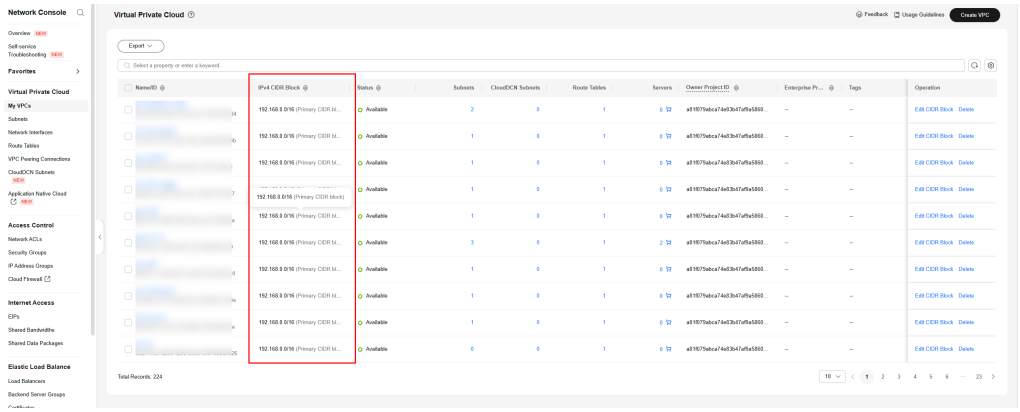
CCI provides image snapshots. You can create an image snapshot using the image to be used, so that the image is not pulled when the workload is started. For details, see **Image Snapshots**.

5 Network Management

How Do I View the VPC CIDR Block?

On the home page of the VPC console, view **Name** and **IPv4 CIDR Block** of VPCs. You can modify the CIDR block of a VPC or create a new one.

Figure 5-1 Viewing the VPC CIDR blocks



Does CCI Support Load Balancing?

CCI supports load balancing. When creating a workload on the CCI console, you can choose to use ELB for both access over the private network or public network. For details, see **Workload Access Through a Private Network Load Balancer**.

CCI works with ELB for load balancing. Generally, load balancers are used to route the traffic from the public network to pods.

After creating a workload on CCI, you can create a Service and configure a load balancer for private or public network access.

1. Public network load balancer: For details, see **Public Network Access**.
2. Private network load balancer: For details, see **Private Network Access**.

How Can a Container Access the Public Network?

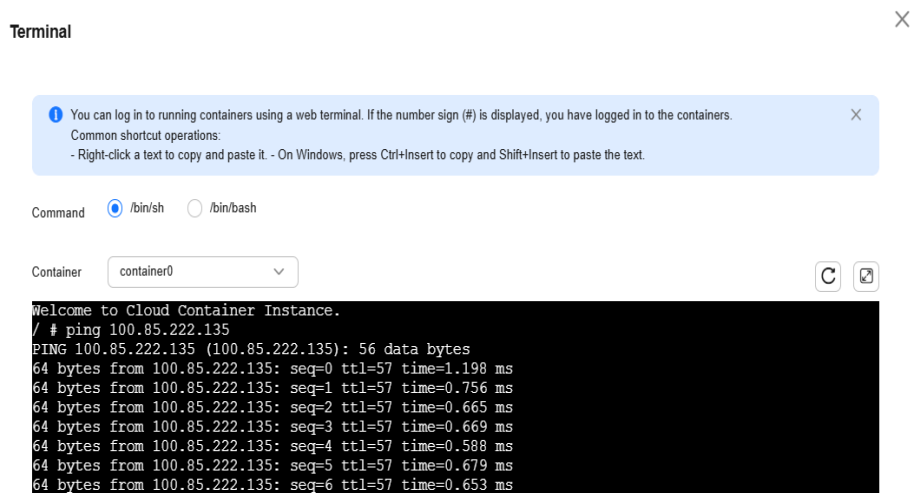
Load balancers can be used to route traffic to pods. If a load balancer has an EIP bound, the pods can be accessed from the public network. For details, see **Public Network Access**.

How Do I Access the Public Network from a Container?

To access the public network from a container, you need to bind an EIP to the workload. The following is an example YAML file:

```
apiVersion: cci/v2
kind: Pod
metadata:
  labels:
    app: nginx
  name: 'nginx'
  annotations:
    yangtse.io/pod-with-eip: 'true'
spec:
  containers:
    - image: xxxxxx
      name: container-0
      resources:
        requests:
          cpu: 500m
          memory: 1Gi
```

After the workload runs normally, you can access the public network from the container.



What Do I Do If Access to a Workload from a Public Network Fails?

1. A workload must be in the running state before it can be accessed from a public network. If the workload is abnormal or not ready, it cannot be accessed from a public network.
2. It may take one to three minutes from the time when the workload is created to the time when it is ready for access from the public network. During this time period, network routes have not yet been configured. As a result, the workload cannot be accessed from the public network.
3. If a workload is inaccessible three minutes after it is created, a possible cause is that the container port is not being listened to. You need to use the image

to check whether the container port is being listened to. If the container port is being listened to, the access failure may be caused by the load balancer. In this case, you need to check the load balancer.

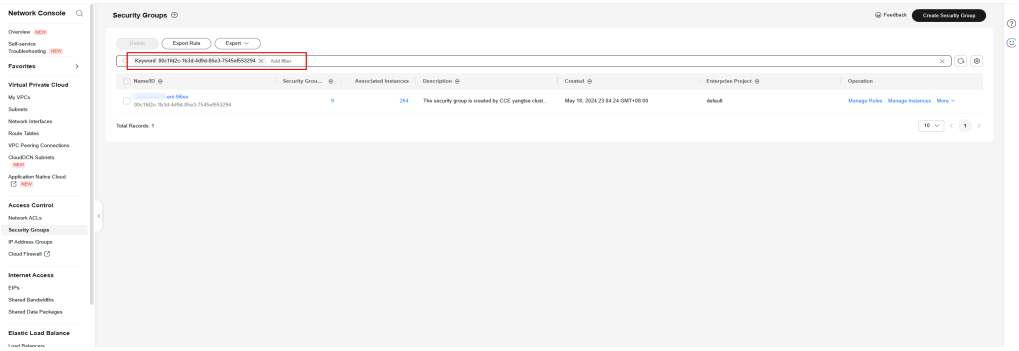
What Do I Do If 504 Is Returned When I Access a Workload?

This is because security group rules do not allow traffic to the pods running in CCI over the listener's port of the load balancer. Check the security group associated with the pods and ensure that security group rules allow traffic to the pods running in CCI over the listener's port of the load balancer.

1. Query the security group associated with the pods by calling the API for querying a network. **spec.securityGroups** in the response is the security group ID.


```
{
    "apiVersion": "yangtse/v2",
    "kind": "Network",
    "metadata": {
        "annotations": {
            "yangtse.io/domain-id": "51ed88507a244b6eb36270c0250fcc96",
            "yangtse.io/project-id": "a81f079abca74e83b47af9a586048b24",
            "yangtse.io/warm-pool-recycle-interval": "24",
            "yangtse.io/warm-pool-size": "10"
        },
        "creationTimestamp": "2024-12-10T09:12:06Z",
        "finalizers": [
            "yangtse.io/network-cleanup"
        ],
        "name": "test-namespace-default-network",
        "namespace": "a81f079abca74e83b47af9a586048b24_test-namespace",
        "resourceVersion": "25163956",
        "uid": "13000c31-2f1d-4f49-9476-569d96b75a48"
    },
    "spec": {
        "networkType": "underlay_neutron",
        "securityGroups": [
            "00c1fd2c-1b3d-4d9d-85e3-7545ef553294",
            "0161da2b-81d6-4dc3-a94d-35ec1b6c486a"
        ],
        "subnets": [
            {
                "subnetID": "14722cef-0ebd-4906-ba3f-46a91840ac2d"
            }
        ]
    }
}
```


2. Log in to the **Network Console** and search for the security group using the obtained security group ID.



3. Click the security group name and add the inbound rules shown in the following figure.

NOTICE

If UDP is used to access the workload from the public network, a rule must be added to allow the ICMP traffic, which is generated by health checks.

Type	Protocol & Port	Source
IPv4	TCP: 80	0.0.0.0/0
IPv4	TCP: 22	0.0.0.0/0
IPv4	TCP: 3389	0.0.0.0/0
IPv4	TCP: 20-21	0.0.0.0/0
IPv4	TCP: 443	0.0.0.0/0
IPv4	ICMP: All	0.0.0.0/0

What Do I Do If the Connection Timed Out?

Symptom

Pods can be created, but the message "[Errno 110] Connection timed out" is displayed when the Python Django SMTP service is used to send emails.

Possible cause

- Only a load balancer was purchased, and no EIPs were purchased. Therefore, containers can only be accessed from the external network. Containers can access the external network after you purchase an EIP for the workload.
- Port 25 is prohibited from sending messages to secure the network environment.

Solutions

- Method 1: Configure an EIP for the workload. The following is an example YAML file:

```
apiVersion: cci/v2
kind: Pod
metadata:
  labels:
    app: nginx
  name: 'nginx'
  annotations:
    yangtse.io/pod-with-eip: 'true'
spec:
  containers:
    - image: xxxxxx
      name: container-0
      resources:
        requests:
          cpu: 500m
          memory: 1Gi
```

- Solution 2: Contact technical support to allow port 25 for the new EIP.

6 Log Collection

Why Are There Lost or Duplicate Logs?

- **Lost logs**

  Cause 1: The log storage duration exceeds the LTS log storage duration.

  Description: CCI collects application logs and reports them to LTS. Logs reported to LTS will be deleted in the early morning of the next day after the log storage duration expires. You can transfer logs to OBS buckets for long-term storage.

  Cause 2: The log file exists for less than 5 seconds.

  Description: CCI detects log files at an interval of 5 seconds. If the duration from the time when a log file is created to the time when it is deleted or renamed is less than 5 seconds, it may not be collected.

- **Duplicate logs**

  Cause 1: The log file is dumped, and the dumped file is still matched.

  Description: If the file name of the configured log path contains a wildcard character, for example, **/tmp/*.log**, after the **/tmp/test.log** file is dumped to **/tmp/test.001.log**, it is still matched by the wildcard rule and is collected again as a new file.

Why Is No Data Displayed During Log Collection?

- Cause 1: **logconf.k8s.io/fluent-bit-log-type: lts** is not configured when TLS is used to collect logs.
- Cause 2: The log stream ID is incorrect.

Why Did the Parameter Verification Fail for Log Collection?

The log stream ID is incorrect.

7 Events

Why Can't Some Pod Events Be Viewed?

CCI events are reported to LTS. Flow control is performed when events are reported. If some pod events are reported too fast, flow control is triggered, which may cause event loss. By default, a maximum of 25 events can be reported by each event source, and event reporting can be resumed in 5 minutes.

8 Accounts

Why Is a Message Displayed Indicating That My Account Is In Arrears Even If It Has Balance?

Symptom

When an account has balance after being reset, CCI displays a message indicating that the account is in arrears during namespace creation.

Solution

Sign out and then sign in again, or clear the browser cache.

Why Is There a Message Indicating That My Account Is Frozen Even It Has Balance?

Symptom

Your account has balance, but CCI displays a message indicating that your account has been frozen due to violation and cannot be used to purchase or use cloud services.

Solution

Submit a service ticket.

Why Can't My Resources Be Deleted?

Symptom

Resources cannot be deleted.

Solution

Some resource permissions are restricted if your account is in arrears. Bring the account current and then delete the resources.

