Cloud Container Instance (CCI 2.0)

Getting Started

Issue	01
Date	2025-09-19

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 Deploying a Static Web Application Using CCI

This section describes how you can use CCI to deploy a static web game application named **2048**.

The following table shows the procedure.

Procedure

Step	Description
Preparations	• Sign up for a HUAWEI ID. • Download ccictl to deploy the static web application.
Step 1: Build an Image and Push It to the SWR Image Repository	Build an image for the application and push the image to the image repository so that the image can be pulled when you create a workload on CCI.
Step 2: Create a Namespace	Create a namespace on CCI for project management.
Step 3: Buy VPC Endpoints	Purchase VPC endpoints to access cloud services that use the network segment starting with 100 over the VPC.
Step 4: Create a Pod	Configure basic information and access information.
Step 5: Access the Pod	Use an EIP to access the workload.
Step 6: Clear Resources	Delete the resources promptly if you no longer need them to avoid additional expenditures.

Preparations

- Before you start, sign up for a HUAWEI ID. For details, see **Signing Up for a HUAWEI ID and Enabling Huawei Cloud Services**.
- You can deploy the static web applications using the console or ccictl. If you use ccictl, download and configure it. For details, see **ccictl Configuration Guide**.

Step 1: Build an Image and Push It to the SWR Image Repository

To deploy an application on CCI, you first need to build an image for the application and push the image to the image repository. Then the image can be pulled when you create a workload on CCI.

Installing the Container Engine

Before pushing an image, you need to install a container engine. Ensure that the container engine version is 1.11.2 or later.

Step 1 Create a Linux ECS with an EIP bound. For details, see **Purchasing an ECS**.

In this example, select **1 vCPUs | 2 GiB** for ECS specifications, **1 Mbit/s** for the bandwidth, and **CentOS 8.2** for the OS.

NOTE

You can also install the container engine on other machines.

Step 2 Go to the ECS list and click **Remote Login** to log in to the ECS.

Step 3 Install the container engine.

```
curl -fsSL get.docker.com -o get-docker.sh
sh get-docker.sh
sudo systemctl daemon-reload
sudo systemctl restart docker
```

----End

Building an Image

The following describes how to use a Dockerfile and the **nginx** image to build the **2048** image. Before building the image, you need to create a Dockerfile.

Step 1 Pull the **nginx** image from the image repository as the base image.

```
docker pull nginx
```

Step 2 Download the **2048** static web application.

```
git clone https://gitee.com/jorgensen/2048.git
```

Step 3 Build a Dockerfile.

1. Run the following command:
   ```
   vi Dockerfile
   ```
2. Edit the Dockerfile.
   ```
   FROM nginx

   MAINTAINER Allen.Li@gmail.com
   COPY 2048 /usr/share/nginx/html
   ```



```
EXPOSE 80

CMD ["nginx", "-g", "daemon off;"]
```

- **nginx** indicates the base image. You can select the base image based on the application type. For example, select a Java image as the base image to create a Java application.
- **/usr/share/nginx/html** indicates the directory for storing the static web.
- **80** indicates the container port.

NOTE

For details about the Dockerfile content and format, see **Dockerfile reference**.

Step 4 Build the **2048** image.

1. Run the following command:
   ```
   docker build -t='2048' .
   ```
   The following information will be displayed after the image build is successful.

   Figure 1-1 Successful image build

   ```
   [root@ecs-8b9a ~]# docker build -t='2048' .
   Sending build context to Docker daemon  1.343MB
   Step 1/5 : FROM nginx
    ---> dd34e67e3371
   Step 2/5 : MAINTAINER Allen.Li@gmail.com
    ---> Running in c10a4ae5b0f2
   Removing intermediate container c10a4ae5b0f2
    ---> bae735c6b9e5
   Step 3/5 : COPY . /usr/share/nginx/html
    ---> 50d8b973a076
   Step 4/5 : EXPOSE 80
    ---> Running in ada8dcf0df3f
   Removing intermediate container ada8dcf0df3f
    ---> 45d7fccc6ad4
   Step 5/5 : CMD ["nginx", "-g", "daemon off;"]
    ---> Running in fcafc573a441
   Removing intermediate container fcafc573a441
    ---> 71eecd989add
   Successfully built 71eecd989add
   Successfully tagged 2048:latest
   ```

2. Query the image.
   ```
   docker images
   ```
   If the following information is displayed, the image has been built.

   Figure 1-2 Querying the image

   ```
   REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
   2048         latest    71eecd989add   3 minutes ago    134MB
   nginx        latest    dd34e67e3371   2 days ago       133MB
   ```

----End

Pushing the Image

Step 1 Access SWR.

1. **Log in to the management console.** In the service list, click **Containers > SoftWare Repository for Container**.
2. In the navigation pane, choose **My Images**. Then click **Upload Through Client**. In the dialog box displayed, click **Temporary Login Command** and click  to copy the temporary login command.

   NOTE

   The temporary login command is valid for 24 hours. To obtain a long-term valid login command, see **Obtaining a Long-Term Valid Login Command**.
3. Run the login command on the server where the container engine is installed.

   The message "login succeeded" will be displayed after a successful login.

Step 2 Push the image.

1. Label the **2048** image on the server where the container engine is installed.

   docker tag [*image-name:image-tag*] [*image-repository-address*]/[*organization-name*]/[*image-name:image-tag*]

   The following is an example:

   docker tag 2048:latest {Image repository address}/cloud-develop/2048:latest

   In the command:

   - **{Image repository address}** indicates the SWR image repository address.
   - **cloud-develop** indicates the organization name of the image.
   - **2048:latest** indicates the image name and tag.
2. Push the image to the image repository.

   docker push [*image-repository-address*]/[*organization-name*]/[*image-name:tag*]

   The following is an example command:

   docker push {Image repository address}/cloud-develop/2048:latest

   If the following information is displayed, the image is pushed to the image repository:

   ```
   6d6b9812c8ae: Pushed
   695da0025de6: Pushed
   fe4c16cbf7a4: Pushed
   v1: digest: sha256:eb7e3bbd8e3040efa71d9c2cacfa12a8e39c6b2ccd15eac12bdc49e0b66cee63 size: 948
   ```

   To view the pushed image, go to the SWR console and refresh the **My Images** page.

----End

Step 2: Create a Namespace

- Create the **webapp** namespace on the console.

Step 1 Log in to the **CCI 2.0 console**.

Step 2 In the navigation pane, choose **Namespaces**.

Step 3 On the **Namespaces** page, click **Create Namespace** in the upper right corner.

Step 4 Configure basic information.

Parameter	Description
Namespace Name	You can create different namespaces for environment isolation. • The name of each namespace must be unique. • Enter 1 to 63 characters starting and ending with a lowercase letter or digit. Only lowercase letters, digits, and hyphens (-) are allowed.
Enterprise Project	Select or create an enterprise project. This parameter is available only for enterprise users who have enabled an enterprise project. After an enterprise project is selected, the security group for the namespace will be created in that project. You can manage namespaces and other resources through the Enterprise Project Management Service (EPS). For more details, see **Enterprise Management**.

Step 5 (Optional) Specify monitoring settings.

Parameter	Description
AOM (Optional)	If this option is enabled, you need to select an AOM instance.

Step 6 Configure the network plane.

Table 1-1 Network plane settings

Parameter	Description
IPv6	If this option is enabled, IPv4/IPv6 dual stack is supported.

Parameter	Description
VPC	Select the VPC where the workloads are running. If no VPC is available, create one first. The VPC cannot be changed once selected. Recommended CIDR blocks: 10.0.0.0/8-22, 172.16.0.0/12-22, and 192.168.0.0/16-22 **NOTICE** • You cannot set the VPC CIDR block and subnet CIDR block to 10.247.0.0/16, because this CIDR block is reserved for workloads. If you select this CIDR block, there may be IP address conflicts, which may result in workload creation failure or service unavailability. If you do not need to access pods through workloads, you can select this CIDR block. • After the namespace is created, you can choose **Namespaces** in the navigation pane and view the VPC and subnet in the **Subnet** column.
Subnet	Select the subnet where the workloads are running. If no subnet is available, create one first. The subnet cannot be changed once selected. • A certain number of IP addresses (**10** by default) in the subnet will be warmed up for the namespace. • You can set the number of IP addresses to be warmed up in **Advanced Settings**. • If warming up IP addresses for the namespace is enabled, the VPC and subnet can only be deleted after the namespace is deleted. **NOTE** Ensure that there are sufficient available IP addresses in the subnet. If IP addresses are insufficient, workload creation will fail.
Security Group	Select a security group. If no security group is available, create one first. The security group cannot be changed once selected.

Step 7 (Optional) Specify advanced settings.

Each namespace provides an IP pool. You can specify the pool size to reduce the duration for assigning IP addresses and speed up the workload creation.

For example, 200 pods are running routinely, and 200 IP addresses are required in the IP pool. During peak hours, the IP pool instantly scales out to provide 65,535 IP addresses. After a specified interval (for example, 23 hours), the IP addresses that exceed the pool size (65535 – 200 = 65335) will be recycled.

Table 1-2 (Optional) Advanced namespace settings

Parameter	Description
IP Pool Warm-up for Namespace	• An IP pool is provided for each namespace, with the number of IP addresses you specify here. IP addresses will be assigned in advance to accelerate workload creation. • An IP pool can contain a maximum of 65,535 IP addresses. • When using general-computing pods, you are advised to configure an appropriate size for the IP pool based on service requirements to accelerate workload startup. • Configure the number of IP addresses to be assigned properly. If the number of IP addresses exceeds the number of available IP addresses in the subnet, other services will be affected.
IP Address Recycling Interval (h)	Pre-assigned IP addresses that become idle can be recycled within the duration you specify here. **NOTE** Recycling mechanism: • Recycling time: The **yangtse.io/warm-pool-recycle-interval** field configured on the network determines when the IP addresses can be recycled. If **yangtse.io/warm-pool-recycle-interval** is set to **24**, the IP addresses can only be recycled 24 hours later. • Recycling rate: A maximum of 50 IP addresses can be recycled at a time. This prevents IP addresses from being repeatedly assigned or released due to fast or frequent recycling.

Step 8 Click **OK**.

You can view the VPC and subnet on the namespace details page.

By default, CCI creates an agency for users to access peripheral services in the namespace. This agency is encrypted and stored in **aksk-secret**. The encryption and decryption material is stored in **system-preset-aeskey**. The two resources are used by CCI and have been hidden on the console. You can call APIs to view them, and you are advised not to configure them.

----End

- Use ccictl to create the **webapp** namespace and the corresponding network. After ccictl is configured, take the following steps:

Step 1 Create a namespace.

```
ccictl create namespace webapp
```

```
root@                                   :~# ccictl create namespace webapp
namespace.cci/                         /webapp created
```

Step 2 Create a network for the namespace. The following is an example YAML file:


```
apiVersion: yangtse/v2
kind: Network
metadata:
  annotations:
    yangtse.io/domain-id: <domain_id> # Account ID
    yangtse.io/project-id: <project_id> # Project ID
  name: cci-network
  namespace: webapp
spec:
  networkType: underlay_neutron
  securityGroups:
    - <security_group_id> # Security group ID
  subnets:
    - subnetID: <subnet_id> # Subnet ID
```

```
root@                                        :~# ccictl apply -f network.yaml
network.yangtse/cci-network created
```

----End

Step 3: Buy VPC Endpoints

VPC endpoints are required for accessing cloud services that use the network segment starting with 100.

- To pull images from a repository of SWR Enterprise Edition, you need a VPC endpoint for accessing OBS.
- To pull images from an SWR public image repository, you need a VPC endpoint for accessing SWR and a VPC endpoint for accessing OBS in the VPC where the workload is deployed.

Step 1 Go to the **VPC endpoint list** page.

Step 2 Buy a VPC endpoint for accessing SWR.

1. On the **VPC Endpoints** page, click **Buy VPC Endpoint**.

   The **Buy VPC Endpoint** page is displayed.

2. Configure the parameters.
   - Select the region configured in **Step 2: Create a Namespace**.
   - Set **Service Category** to **Cloud service**, search for **swr**, and select the VPC endpoint service for accessing SWR.
   - **VPC**: Select the VPC configured in **Step 2: Create a Namespace**.
   - **Subnet**: Select the subnet configured in **Step 2: Create a Namespace**.

   Specify other parameters as needed.

Figure 1-3 Buying a VPC endpoint for accessing SWR

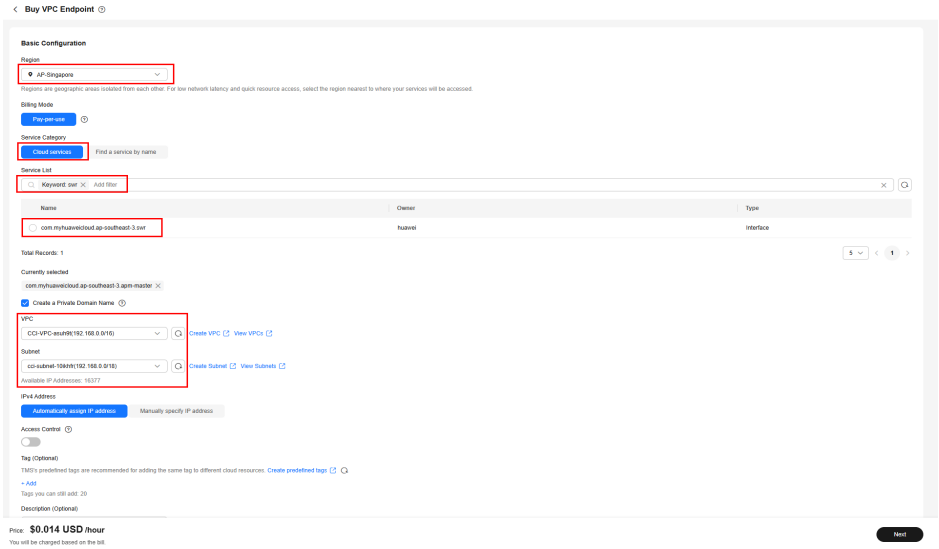


Table 1-3 VPC endpoint parameters

Parameter	Example	Description
Region	CN-Hong Kong	Specifies the region where the VPC endpoint will be used to connect a VPC endpoint service. Resources in different regions cannot communicate with each other over a private network. For lower latency and quicker access, select the region nearest to your on-premises data center.
Billing Mode	Pay-per-use	Specifies the billing mode of the VPC endpoint. VPC endpoints can be used or deleted at any time. VPC endpoints support only pay-per-use billing based on the usage duration.
Service Category	Cloud services	Select **Cloud services** when you buy a VPC endpoint for accessing SWR.
Service List	-	This parameter is available only when you select **Cloud services** for **Service Category**. VPC endpoint services have been created. You only need to select the desired one.
VPC	-	Select the VPC configured in **Step 2: Create a Namespace**.

Parameter	Example	Description
Subnet	-	Specifies the subnet where the VPC endpoint is to be deployed.
Route Table	-	This parameter is available only when you create a VPC endpoint for connecting to a gateway VPC endpoint service. **NOTE** This parameter is available only in the regions where the route table function is enabled. You are advised to select all route tables. Otherwise, access to the VPC endpoint service of the gateway type may fail. Select the route tables required for the VPC where the VPC endpoint is to be deployed. For details about how to add a route, see **Adding Routes to a Route Table** in the *Virtual Private Cloud User Guide*.
Policy	-	Specifies the VPC endpoint policy. VPC endpoint policies are a type of resource-based policies. You can configure a policy to control which principals can use the VPC endpoint to access VPC endpoint services.
Tag	example_key1 example_value1	Specifies the tag that is used to classify and identify the VPC endpoint. The tag settings can be modified after the VPC endpoint is purchased
Description	-	Provides supplementary information about the VPC endpoint.

Table 1-4 Tag requirements for VPC endpoints

Parameter	Requirement
Tag key	– Cannot be left blank. – Must be unique for each resource. – Can contain a maximum of 128 characters. – Cannot start or end with a space or contain special characters =*<>\,|/ – A tag key can contain letters, digits, spaces, and any of the following characters: _.:=+-@. It cannot start or end with a space, or start with _sys_.
Tag value	– Can be left blank. – Can contain a maximum of 255 characters. – Cannot start or end with a space or contain special characters =*<>\,|/ – A tag value can contain letters, digits, spaces, and characters _.:/=+-@. It cannot start or end with a space.

3. Confirm the settings and click **Next**.
   - If the configuration is correct, click **Submit**.
   - If any parameter is incorrect, click **Previous** to modify it as needed and then click **Submit**.

Step 3 Buy a VPC endpoint for accessing OBS.

1. On the **VPC Endpoints** page, click **Buy VPC Endpoint**.

   The **Buy VPC Endpoint** page is displayed.
2. Configure the parameters.
   - Select the region configured in **Step 2: Create a Namespace**.
   - Set **Service Category** to **Find a service by name**. You can obtain the name of the VPC endpoint service for OBS by submitting a service ticket. Enter the service name and click **Verify** to confirm that the service name is correct.
   - **VPC**: Select the VPC configured in **Step 2: Create a Namespace**. When you create a VPC endpoint for a VPC endpoint service of the gateway type, **Route Table** is displayed. You are advised to select all route tables. Otherwise, the network may be unreachable.

   Specify other parameters as needed.

Figure 1-4 Buying a VPC endpoint for accessing OBS

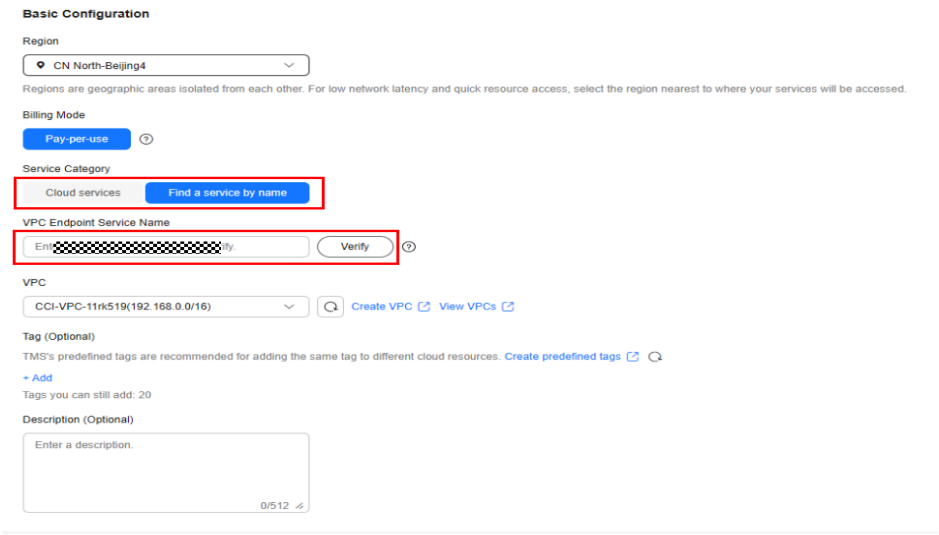


Table 1-5 VPC endpoint parameters

Parameter	Example	Description
Region	CN-Hong Kong	Specifies the region where the VPC endpoint will be used to connect a VPC endpoint service. Resources in different regions cannot communicate with each other over a private network. For lower latency and quicker access, select the region nearest to your on-premises data center.
Billing Mode	Pay-per-use	Specifies the billing mode of the VPC endpoint. VPC endpoints can be used or deleted at any time. VPC endpoints support only pay-per-use billing based on the usage duration.
Service Category	Find a service by name	Select **Find a service by name** when you buy a VPC endpoint for accessing OBS.
VPC	-	Select the VPC configured in **Step 2: Create a Namespace**.
Subnet	-	Specifies the subnet where the VPC endpoint is to be deployed.

Parameter	Example	Description
Route Table	-	This parameter is available only when you create a VPC endpoint for connecting to a gateway VPC endpoint service. **NOTE** This parameter is available only in the regions where the route table function is enabled. You are advised to select all route tables. Otherwise, access to the VPC endpoint service of the gateway type may fail. Select the route tables required for the VPC where the VPC endpoint is to be deployed. For details about how to add a route, see **Adding Routes to a Route Table** in the *Virtual Private Cloud User Guide*.
Policy	-	Specifies the VPC endpoint policy. VPC endpoint policies are a type of resource-based policies. You can configure a policy to control which principals can use the VPC endpoint to access VPC endpoint services.
Tag	example_key1 example_value1	Specifies the tag that is used to classify and identify the VPC endpoint. The tag settings can be modified after the VPC endpoint is purchased
Description	-	Provides supplementary information about the VPC endpoint.

Table 1-6 Tag requirements for VPC endpoints

Parameter	Requirement
Tag key	– Cannot be left blank. – Must be unique for each resource. – Can contain a maximum of 128 characters. – Cannot start or end with a space or contain special characters =*<>\,|/ – A tag key can contain letters, digits, spaces, and any of the following characters: _.:=+-@. It cannot start or end with a space, or start with _sys_.
Tag value	– Can be left blank. – Can contain a maximum of 255 characters. – Cannot start or end with a space or contain special characters =*<>\,|/ – A tag value can contain letters, digits, spaces, and characters _.:/=+-@. It cannot start or end with a space.

3. Confirm the settings and click **Next**.
   - If the configuration is correct, click **Submit**.
   - If any parameter is incorrect, click **Previous** to modify it as needed and then click **Submit**.

Step 4 Click **Back to VPC Endpoint List** after the task is submitted.

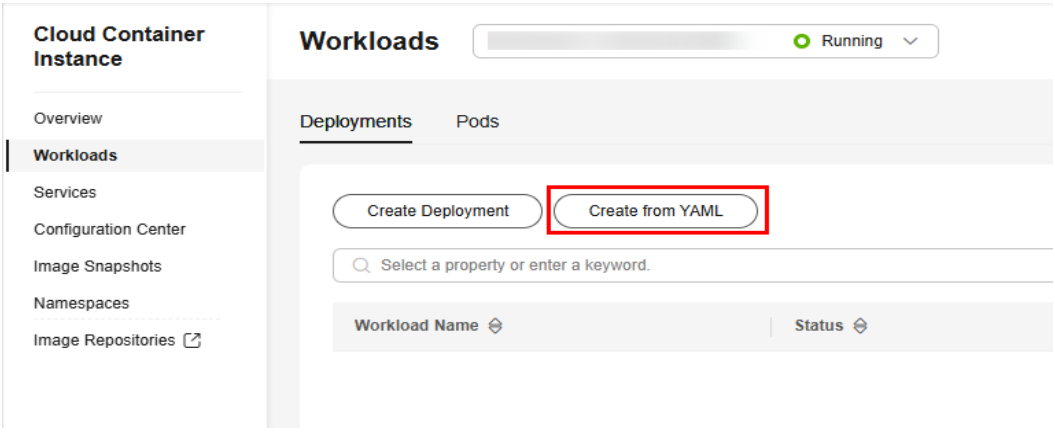
Step 5 View the endpoint details by clicking each endpoint ID.

----End

Step 4: Create a Pod

Before creating a pod, use VPC Endpoint to connect your namespace to the networks of other cloud services. For details, see **Purchasing VPC Endpoints**. After VPC endpoints are ready, you can perform the following operations to create a pod:

- Create a pod on the console.
  a. Log in to the **CCI 2.0 console**.
  b. In the navigation pane, choose **Workloads**. On the **Pods** tab, click **Create from YAML**.



c. Specify basic information. The following is an example YAML file:

```
apiVersion: cci/v2
kind: Pod
metadata:
  labels:
    app: webapp-2048
  name: webapp-2048
  namespace: webapp
spec:
  containers:
    - image: {Image repository address}/cloud-develop/2048:latest # Uploaded image
      name: webapp-2048
      ports:
        - containerPort: 80
          protocol: TCP
      resources:
        limits:
          cpu: 500m
          memory: 1Gi
        requests:
          cpu: 500m
          memory: 1Gi
  dnsPolicy: Default
```

- Use cctl to create a pod. Save the YAML file as **pod.yaml** and run the following command:

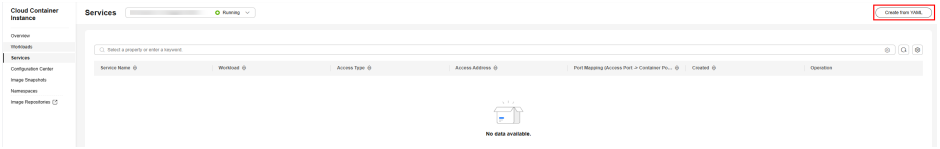
```
cctl apply -f pod.yaml
```

```
root@                                    :~# cctl apply -f pod.yaml
pod.cci/webapp-2048 created
```

Step 5: Access the Pod

- Create a Service for accessing the pod on the console.

  a. In the navigation pane, choose **Services**. On the displayed page, click **Create from YAML** on the right.



  b. Create a Service of the LoadBalancer type. The load balancer must have an EIP. The following is an example YAML file:

```
kind: Service
apiVersion: cci/v2
metadata:
  name: service-2048
```



```
  namespace: webapp
  annotations:
    kubernetes.io/elb.class: elb
    kubernetes.io/elb.id: <elb_id> # The load balancer must have an EIP.
spec:
  ports:
    - name: service-port
      protocol: TCP
      port: 80
      targetPort: 80
  selector:
    app: webapp-2048
  type: LoadBalancer
```

- Use cctl to create a Service. Save the preceding YAML file as **service.yaml** and run the following command:

```
cctl apply -f service.yaml
```

```
root@                                   :~# cctl apply -f service.yaml
service.cci/service-2048 created
```

After the pod and Service are created, you can enter **http://*<EIP-of-load-balancer>*/2048/index.html** in the address box of your browser to access your web application.

Step 6: Clear Resources

- On the CCI console, perform the following operations to clear resources:
  a. Log in to the **CCI 2.0 console**.
  b. In the navigation pane, choose **Workloads**. Then click the **Pods** tab.
  c. Locate the pod to be deleted and click **Delete** in the **Operation** column.

  NOTE

  To delete the load balancer used by a Service, delete the Service on the CCI 2.0 console, and then delete the load balancer on the ELB console.

- Use cctl to clear resources.

```
cctl delete -f service.yaml
cctl delete -f pod.yaml
cctl delete namespace webapp
```

```
root@                                   :~# cctl apply -f service.yaml
service.cci/service-2048 created
root@                                   :~# cctl delete -f service.yaml
service.cci "service-2048" deleted
root@                                   :~# cctl delete -f pod.yaml
pod.cci "webapp-2048" deleted
root@                                   :~# cctl delete namespace webapp
namespace.cci "webapp" deleted
```

