

# Tencent Kubernetes Engine

## TKE Edge Cluster Guide

### Product Documentation



## Copyright Notice

©2013-2022 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

## Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

## Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

# Contents

## TKE Edge Cluster Guide

Overview

### TKE Edge Cluster Management

Creating a Cluster

Node Pool Management

Edge Node Pool (NodeUnit)

Node Management

Connecting to a Cluster

ENI

Cluster Resource Quota Adjustment

### Kubernetes Object Management

Workload Management

Service Management

# TKE Edge Cluster Guide

## Overview

Last updated : 2022-10-14 10:49:28

## Product Introduction

Edge Cloud Kubernetes Engine (ECK) is a container system for managing edge cloud resources from a centralized cloud. ECK is fully compatible with native Kubernetes. You can manage nodes in multiple data centers with one cluster and deliver applications to all edge servers with one click. ECK also comes with edge autonomy and distributed health check features.

## Concepts

### Containers and images

Containers are lightweight virtualization tools applied at the system level. With the ability to isolate and control system resources, containers restrict global resources access to processes in selected containers. A container image is a virtual machine snapshot and can be seen as the static form of a container. An image defines all files and dependencies required to run a container, ensuring consistency for running the container.

A container packages an application and its dependencies into an image, and then uses the image to generate a resource-isolated environment to run the application. This allows the application to run independently in a consistent environment in a simple and efficient manner.

### Kubernetes

Kubernetes is an open-source Container Orchestration Engine (COE) inspired by a Google project called Borg. It is one of the most important components of the Cloud Native Computing Foundation (CNCF). Kubernetes provides production-level features such as application orchestration, container scheduling, service discovery, and autoscaling. For more information, see [Kubernetes Documentation](#).

## Benefits of ECK

### Native support

ECK is an out-of-the-box service that supports the latest Kubernetes version and native Kubernetes cluster management methods.

### **Availability across data centers**

ECK is a Kubernetes service with master components hosted in the cloud and worker nodes located anywhere you want. Users do not need to provide resources required by the master components.

### **Security and reliability**

ECK supports private and public network certificates separation and minimal node permissions to avoid cluster access leakage. TLS encryption is used in communication between the cloud and the edge to protect system management data from leakage and tampering.

### **System disaster recovery**

ECK provides a reliable edge autonomy capability for cloud-edge communication scenarios. It also supports distributed cluster health check to help you determine the best time to migrate pods.

### **Easy system OPS**

ECK leveraged Tencent's years of experience in deep tunneling technologies to enable admins to log in to containers on edge servers directly from the cloud, even if the edge devices do not have public IP addresses.

### **Management across clouds**

ECK supports management of public cloud, private cloud, Tencent Cloud, and any other cloud computing resources.

## Pricing

ECK is a Kubernetes service with master components hosted. Like TKE, ECK does not charge fees for management resources such as hosted control plane and Etcd.

Computing nodes are provided by users and therefore do not generate any additional charges.

## Use Cases

### **Edge computing**

ECK helps you manage edge computing resources, allowing you to assign and schedule resources, deploy, upgrade and terminate applications, and perform system OPS from the cloud.

### **Management across clouds**

With ECK, you can easily manage computing resources stored in various locations, from different cloud providers to your on-premises data centers, enjoying the convenience of centralized cloud management.

## Additional Services

For more information on how to call TencentCloud APIs to access Tencent Cloud products and services, see [TencentCloud APIs](#).

# TKE Edge Cluster Management

## Creating a Cluster

Last updated : 2020-04-26 18:12:44

### Scenario

This document describes how to create an edge cluster to use TKE Edge from the Tencent Cloud TKE console.

### Prerequisites

- To use TKE Edge, [submit a ticket](#) to apply for it.
- Log in to the [CAM console](#) to activate the required permissions.

### Notes on the Container Network

Edge TKE uses the node-side network to build the overlay network. Therefore, ensure that the cluster network and container network do not conflict with the node network on the edge server side.

### Directions

1. Log in to the [Tencent Cloud TKE console](#) and click **Edge Clusters** in the left sidebar.
2. On the "Edge Clusters" page, click **Create** to go to the "Create Edge Cluster" page.
3. On the "Create Edge Cluster" page, create an edge cluster based on the following information.
  - **Cluster name:** indicates the name of the edge cluster to be created, with a maximum length of 60 characters.
  - **Kubernetes version:** Kubernetes version 1.16 is currently supported. This version will be updated when a newer Kubernetes version is published by the Kubernetes community.
  - **Region:** select the region that is closest to your location to minimize access latency and improve the download speed.
  - **Cluster network:** assign a network for the cluster according to the internal network management of edge servers.
  - **Pod CIDR:** you need to assign a container network for the cluster according to the internal network management of edge servers. Therefore, plan the cluster size in advance to assign an IP range with sufficient IP addresses for

the container network. **The pod CIDR block cannot overlap with IP ranges used by a VPC instance and existing Kubernetes clusters in the VPC instance. In addition, it cannot be modified once created.**

- **Service CIDR:** you need to assign a service network for the cluster according to the internal network management of edge servers. Therefore, plan the cluster size in advance to assign an IP range with sufficient IP addresses for the service network. **The service CIDR block cannot overlap with IP ranges used by a VPC instance and existing Kubernetes clusters in the VPC instance. In addition, it cannot be modified once created.**
- **Cluster description:** indicates information about the cluster, which is displayed on the **Cluster Information** page.

4. Click **Done** to finish creating the Master components of the cluster. You can check the progress of cluster creation on the "Edge Clusters" page.

## Next Steps

Go to "Node Management" and add nodes to the created edge cluster.



# Node Pool Management

## Edge Node Pool (NodeUnit)

Last updated : 2022-10-12 11:37:13

## Overview

This document describes how to manage edge node pools of an edge container in the TKE console.

This update reconstructs the UI interaction between the earlier versions of NodeGroup and NodeUnit. Clusters created after March 29, 2022 will use the new interaction logic, and clusters on earlier versions will not be affected. In this update, the edge node pool corresponds to the design of NodeUnit in SuperEdge, and node pool category corresponds to the design of NodeGroup in SuperEdge. For detailed design principles, see [Managing Edge Resources by Application Resource Pool](#).

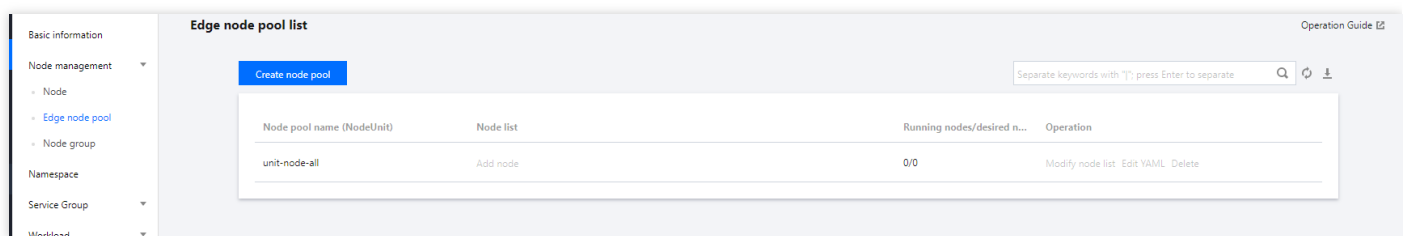
## Directions

Note :

As operations on an edge node pool will affect the labels of the nodes, perform such operations with caution.

### Creating an edge node pool

1. Log in to the [TKE console](#) and select **Edge Clusters** on the left sidebar.
2. Click a **Cluster ID** to enter the cluster details page.
3. Select **Node Management > Edge Node Pool** on the left sidebar to enter the **Edge Node Pool List** page.



Each cluster has a default node pool ( `unit-node-all` ), which contains all edge nodes added to the cluster.

4. Click **Create Node Pool**. On the **Create Edge Node Pool** page, enter the node pool name and add existing nodes as needed.

### Create edge node pool ✕

Node pool name

The name can contain up to 60 characters. It supports lower-case letters, digits and "-", and must start with a letter and end with a digit or letter.

Node list  ↻

5. Click **Done**. You can view added nodes in the edge node pool list.

Edge node pool list				
Node pool name (NodeUnit)	Node list	Running nodes/desired n...	Operation	
unit-node-all	<a href="#">Add node</a>	0/0	<a href="#">Modify node list</a> <a href="#">Edit YAML</a> <a href="#">Delete</a>	
wsd12sfss	<a href="#">Add node</a>	0/0	<a href="#">Modify node list</a> <a href="#">Edit YAML</a> <a href="#">Delete</a>	

## Managing an edge node pool

1. Log in to the [TKE console](#) and select **Edge Clusters** on the left sidebar.
2. Click a **Cluster ID** to enter the cluster details page.
3. Select **Node Management > Edge Node Pool** on the left sidebar to enter the **Edge Node Pool List** page.
4. Click **Modify Node List** on the right of an edge node pool. On the **Modify Edge Node Pool** page, you can add and delete existing nodes.

## Deleting an edge node pool

1. Log in to the [TKE console](#) and select **Edge Clusters** on the left sidebar.
2. Click a **Cluster ID** to enter the cluster details page.
3. Select **Node Management > Edge Node Pool** on the left sidebar to enter the **Edge Node Pool List** page.
4. Click **Delete** on the right of an edge node pool to delete the pool.

# Node Management

Last updated : 2020-11-24 17:56:15

## Scenario


This document describes how to add nodes to the edge cluster that you created.

## Prerequisites

Prepare as instructed below:

- Node source: you can use an existing server in the [CVM console](#) or the [ECM console](#), or a server on another platform or in your on-premises data center.
- Node processor: supports x86\_64, ARM, and ARM64.
- The following operating systems of nodes are supported:
  - Ubuntu 18.04 and 16.04
  - CentOS 7.6, 7.5, and 7.4
  - Tencent Linux Release 2.4 and 2.2 (Final)
  - SUSE Linux Enterprise Server 12 SP3
  - Debian 9.0
- Ensure that `wget`, `systemctl`, and `iptables` have been installed on the node to be added.
- The node network must be able to actively access the internet.

## Directions

1. Log in to the [Tencent Cloud TKE console](#) and click **Edge Clusters** in the left sidebar.
2. On the **Edge Clusters** page, click the cluster ID to go to the **Deployment** management page.
3. In the left sidebar, choose **Node Management** > **Node** to go to the **Node List** page.
4. On the **Node List** page, click **Add a node**.
5. In the **Add a node** window that appears, complete the following steps to obtain the node initialization script.
  - i. Obtain the initial configuration from the **Configure** step. You can modify the following parameters:
    - **Interface**: indicates the network interface used by the node to communicate within the private network.
    - **nodeName**: indicates the name of the node in the cluster. This name must be unique in the cluster.
  - ii. Click **Next: install**. If you are prompted to enable public network access, click  to enable it.
  - iii. In the **Install** step, copy the script download command for downloading the node initialization script.

6. Log in to the prepared server, switch to the `root` account, and run the copied command.
7. Run the following commands to execute the node initialization script.


```
chmod +x script.sh
```

```
./script.sh
```

8. After the script is successfully executed, go to the "Node List" page and refresh it to check that the newly added node exists.  
You can also perform other node operations, such as draining, removing, cordoning, or uncordoning a node, and editing the node label.

## Relevant Operations

### Disabling public network access for a cluster

1. On the **Edge Clusters** page, click **View cluster credentials** for the cluster for which you want to disable public network access.
2. In the "Cluster Credentials" window that appears, click  for "public network access" to disable public network access for the cluster.

# Connecting to a Cluster

Last updated : 2020-04-26 18:12:44

## Scenario

This document describes how to connect a local client to an edge cluster through kubectl, which is the Kubernetes command-line tool.

## Prerequisites

- The cURL software program has been installed.
- Select the appropriate way to obtain kubectl based on the operating system:

Replace `v1.8.13` in the command with the kubectl version required by your business.

- **MacOS X**

Run the following command to obtain kubectl:

```
curl -LO https://storage.googleapis.com/kubernetes-release/release/v1.8.13/bin/darwin/amd64/kubectl
```

- **Linux**

Run the following command to obtain kubectl:

```
curl -LO https://storage.googleapis.com/kubernetes-release/release/v1.8.13/bin/linux/amd64/kubectl
```

- **Windows**

Run the following command to obtain kubectl:

```
curl -LO https://storage.googleapis.com/kubernetes-release/release/v1.8.13/bin/windows/amd64/kubectl.exe
```

## Directions

### Installing kubectl

1. Install kubectl as instructed in [Installing and Setting Up kubectl](#).

- If you have already installed kubectl, skip this step.
- This step uses the Linux operating system as an example.

2. Run the following commands to grant permissions to use kubectl.

```
chmod +x ./kubectl  
  
sudo mv ./kubectl /usr/local/bin/kubectl
```

3. Run the following command to check the installation result.

```
kubectl version
```

If the output is similar to the following version information, the installation was successful.

```
Client Version: version.Info{Major:"1", Minor:"5", GitVersion:"v1.5.2", GitComm  
it:"08e099554f3c31f6e6f07b448ab3ed78d0520507", GitTreeState:"clean", BuildDat  
e:"2017-01-12T04:57:25Z", GoVersion:"go1.7.4", Compiler:"gc", Platform:"linux/a  
md64"}
```

## Obtaining cluster certificate information

1. Log in to the [Tencent Cloud TKE console](#) and click **Edge Clusters** in the left sidebar.
2. On the **Edge Clusters** page, click **View the cluster credential** for the cluster to be connected.
3. In the **Cluster Credential** window that appears, you can view, copy, and download the credential.

You can save the cluster access credential locally by clicking **Copy** or **Download** as needed.

4. In **Internet access**, click  to enable internet access for the cluster. You can also see [Configuring kubectl autocomplete](#) for accessing with the cluster access credential.

## Using kubectl to manipulate a cluster through certificate information

### Request method

The kubectl command format is as follows:

```
--kubeconfig=<Local cluster access credential>
```

## Example

Run the following command to view existing namespaces under the cluster.

```
kubectl get namespace --kubeconfig=cls-8ipgf8u4.kubeconfig
```

The cluster credential used in this example is `cls-8ipgf8u4.kubeconfig`. In real-life cases, replace it with the actual credential.

If a message similar to the following is returned, the request was successful.

```
NAME STATUS AGE
default Active 11d
kube-system Active 11d
```

## Configuring kubectl autocomplete

You can configure kubectl autocomplete to improve usability by running the following command.

```
source <(kubectl completion bash)
```

# ENI

Last updated : 2022-08-26 17:44:48

## Overview

This document describes how to enable and bind ENI to Pods on CVM edge nodes to implement a high-availability network scheme.

- [Enabling ENI](#)
- [Disabling ENI](#)

## Directions

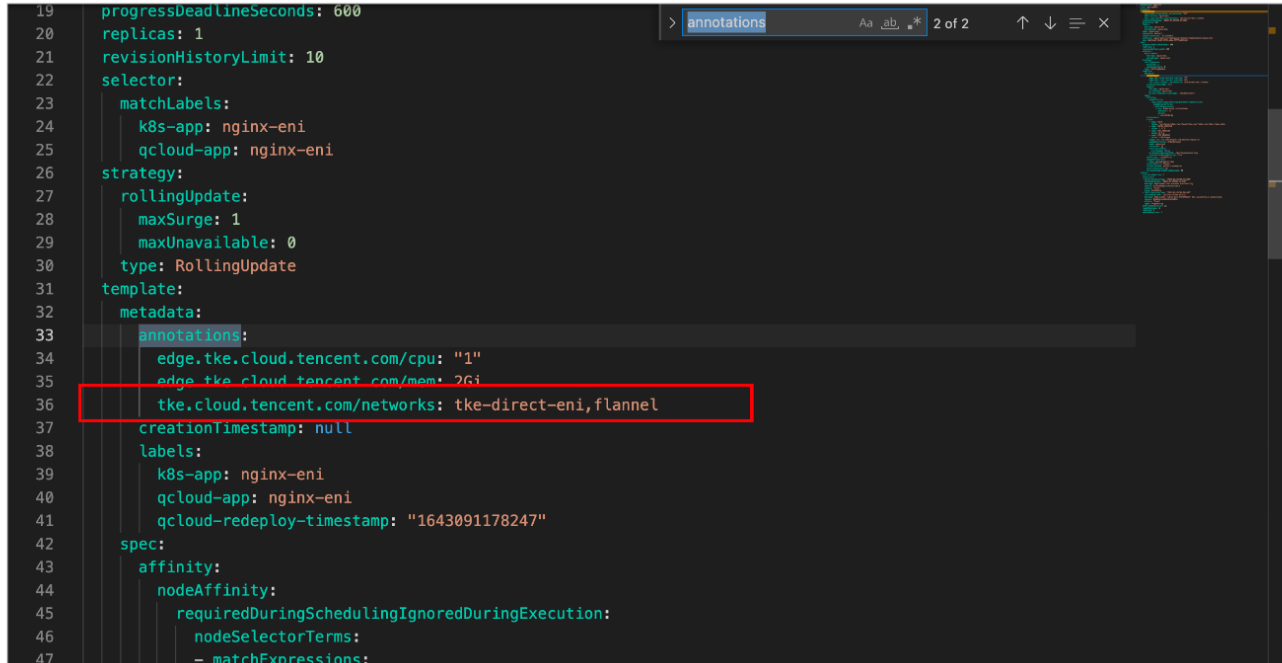
### Enabling ENI

1. Log in to the [TKE console](#) and select **Edge Clusters** on the left sidebar.
2. Click the ID of the target cluster to enter its details page.
3. Select **Basic Information** on the left to enter the **Basic Information** page of the cluster and toggle on the **Enable ENI** switch. The detailed directions are as follows:
  - 3.1 Click the **Access API key** hyperlink to enter the key information page.
  - 3.2 Copy the `ID` and `Key` , return to the **Confirm Key Information** pop-up window, and enter the information.

```
![] (https://qcloudimg.tencent-cloud.cn/raw/ca8a929b58a5a2a71cf9ac5296682ec8.png)
```
  - 3.3 Click **OK**.
4. Select **Workload > Deployment** on the left to enter the Deployment list page. If Deployments exist in the list, skip this step; otherwise, create a Deployment as instructed in [Deployment Management](#).
5. Select **Node Management > Node** on the left to enter the node list page. If CVM nodes exist in the list, skip this step; otherwise, create a CVM node as instructed in [Node Management](#).
6. Configure the ENI in the Pods in the edge cluster.



- The configuration is as follows:



```
19 progressDeadlineSeconds: 600
20 replicas: 1
21 revisionHistoryLimit: 10
22 selector:
23   matchLabels:
24     k8s-app: nginx-eni
25     qcloud-app: nginx-eni
26 strategy:
27   rollingUpdate:
28     maxSurge: 1
29     maxUnavailable: 0
30   type: RollingUpdate
31 template:
32   metadata:
33     annotations:
34       edge.tke.cloud.tencent.com/cpu: "1"
35       edge.tke.cloud.tencent.com/mem: "2Gi"
36       tke.cloud.tencent.com/networks: tke-direct-eni,flannel
37   creationTimestamp: null
38   labels:
39     k8s-app: nginx-eni
40     qcloud-app: nginx-eni
41     qcloud-redeploy-timestamp: "1643091178247"
42 spec:
43   affinity:
44     nodeAffinity:
45       requiredDuringSchedulingIgnoredDuringExecution:
46         nodeSelectorTerms:
47           - matchExpressions:
```

- The ENI capability of edge clusters is supported only by Tencent Cloud CVM node resources. Therefore, when deploying an application, you need to use the `nodeAffinity` capability to schedule the Pod mounted with the ENI to the real CVM edge node (you can enter multiple CVM node IDs).
- Below is the actual code:

```
template:
metadata:
annotations:
tke.cloud.tencent.com/networks: tke-direct-eni,flannel

spec:
affinity:
nodeAffinity:
requiredDuringSchedulingIgnoredDuringExecution:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
values:
- cvm-2cxgi4ow # CVM node ID of the access target
```

## Disabling ENI

1. Log in to the [TKE console](#) and select **Edge Clusters** on the left sidebar.
2. Click the ID of the target cluster to enter its details page.
3. Select **Basic Information** on the left to enter the **Basic Information** page of the cluster and toggle off the **Enable ENI** switch.

# Cluster Resource Quota Adjustment

Last updated : 2022-11-02 11:43:23

TKE Edge automatically applies a set of resource quotas to namespaces in clusters with no more than five nodes ( $0 < \text{nodeNum} \leq 5$ ) and clusters with more than five and fewer than 20 nodes ( $5 < \text{nodeNum} < 20$ ). You cannot adjust the quotas as they will protect the cluster control plane from instability caused by potential bugs in an application after it is deployed in the cluster.

You can run the following command to check the quotas:

```
kubectl get resourcequota tke-default-quota -o yaml
```

Note :

To check the `tke-default-quota` object of a specified namespace, add `--namespace` to specify the namespace.

To adjust the quotas in special scenarios, [submit a ticket](#) for application.

# Kubernetes Object Management

## Workload Management

Last updated : 2020-04-26 18:12:45

### Scenario

This document describes how to run your service through various workloads in an edge cluster.

### Prerequisites

- You have an edge cluster that is in the Running state. For more information, see [Creating a Cluster](#).
- The cluster has an appropriate namespace that is in the Active state.
- The cluster has sufficient edge computing resources to run the workloads.

### Workload Types

#### Deployment

A Deployment declares a template for a pod and a policy for controlling how the pod runs. It is used to deploy stateless applications. You can specify the number of replicas, scheduling policy, and update policy for a pod running in the Deployment as required.

#### StatefulSet

A StatefulSet is primarily used to manage stateful applications. Pods created will come with a persistent identifier according to the specification. The identifier will not change even if the pod is migrated or restarted after termination. When persistent storage is required, you can use the identifiers to map to the corresponding storage volumes. If the application does not require persistent identifiers, we recommend that you use Deployment to deploy the application.

#### Job

A Job creates one or more pods and ensures that these pods run according to the specified rules until a specified number of them successfully terminate. Jobs can be used in many scenarios, such as batch computing and data analysis. You can specify the required number of completions, concurrency (the number of pods running at any instant), and the restart policy as required.

When a Job is completed, no more pods are created, but existing pods will be retained. You can view the logs of the completed pods in **Logs**. Deleting a Job cleans up the pods it created, and the logs of these pods will be invisible.

## CronJob

A CronJob object resembles one line of a crontab (cron table) file. It runs a Job periodically on a given schedule and uses the Cron format.

The cron format is as follows:

```
# File format description
# --minute (0 - 59)
# | --hour (0 - 23)
# | | --day (1 - 31)
# | | | --month (1 - 12)
# | | | | --day of week (0 - 6)
# | | | | |
# * * * * *
```

## Directions

1. Log in to the [Tencent Cloud TKE console](#) and click **Edge Clusters** in the left sidebar.
2. On the "Edge Clusters" page that appears, click the ID of the cluster where the workload that you want to create is located to go to the "Deployment" page.
3. Click **Create** to go to the "Create Workload" page.
4. On the "Create Workload" page, enter a name for the workload and select the workload type.

For the specific parameter settings for each type of workload, see the following:

- [Deployment Management](#)
- [StatefulSet Management](#)
- [Job Management](#)
- [CronJob Management](#)

# Service Management

Last updated : 2022-04-18 14:36:59

## Scenario

This document describes how to expose workloads for access through Services.

## Prerequisites

- You have an edge cluster that is in the Running state. For more information, see [Creating a Cluster](#).
- The cluster has an appropriate namespace that is in the Active state.

## Service Types

### Service

A Service defines a policy for gaining access to backend pods and provides a static virtual IP address for access. You can gain access to backend pods in load balancing mode through the Service.

Services support the following types:

- ClusterIP: provides an entry that is reachable by other Services or containers within the cluster. This type supports the TCP/UDP protocol. For example, it can be utilized by MySQL services to ensure network isolation. For more information, see [Kubernetes Service](#).
- NodePort: provides an access method by mapping a node port to a container. This type supports the TCP/UDP protocol. It is useful when you want to load-balance your businesses to nodes. For more information, see [Kubernetes Service](#).

## Directions

For Service-related operations, see [Service Management](#).

Note :

Intra-cluster access of edge clusters only supports Kubernetes native access methods. To use VPC instances on the computing resource side, you need to integrate the VPC solutions of your cloud providers.

