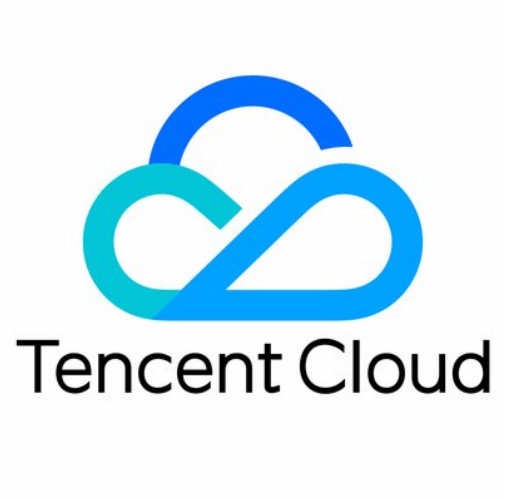
# Tencent Kubernetes Engine Services

# Product Introduction



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

Services
- Basic Operations of Services
- Setting Limit on Service Resources
- Setting Access Mode of Services

# Services
# Basic Operations of Services

Last updated：2018-07-05 15:56:53

## Creating Service

1. Log in to the Tencent Cloud TKE console.
2. Click **Service** in the left navigation bar, and click **New** in the service list page.


3. Configure basic service information.
   - **Service Name**: The name of the service to be created, which has a length limited to 63 characters comprised of lowercase letters, numbers and "-". It starts with a lowercase letter and ends with a lowercase letter or a number.
   - **Region**: Select the region where the service is deployed.
   - **Cluster**: Select a cluster to run your service. You need to select a running cluster with available CVMs in it.
   - **Service description**: Information of created service. This information is displayed in **Service Information** page.


4. Configure volume.

   Click **Add Volume** when you specify a specific path to which a container is mounted.

   > **Note:**
   > If no source path is specified, a temporary path is assigned by default.

   - **Type**: Four types of volumes are supported: local disk, cloud disk, NFS disk, and configuration file. For more information, please see How to Use TKE Volume.
   - **Name**: Volume name.
   - **Path**: Specify the path to which a container is mounted.

5. Configure a container.
   - **Name**: The name of the container to be created, with a length limited to 63 characters.
   - **Image**: Click **Select Image** to create a service under My Images, My Favorites, TencentHub Image, DockerHub Image or other images.
   - **Tag**: The default image tag for TKE. If you need to use a different image tag, click tag display window to select one.


6. Other settings
   - **Number of pods**: A pod consists of one or more relevant containers. You can specify the number of pods by clicking + or -.
   - **Service access method**: The method for accessing a service determines the network attribute of this service. Different access methods offer different network capabilities. For more information about the

four access methods, please see Configuration of Service Access Methods.


7. Click **Create Service** to complete the creation process.

# Updating the Number of Pods

1. Click **Services** in the left navigation bar of TKE console to enter the service list, and click **Update Number of Pods**.

2. You can specify the number of pods by clicking + or -. Click **OK** after the configuration is completed.

# Updating Service

1. Click **Services** in the left navigation bar of TKE console to enter the service list, and click **Update Service**.


2. Click **Start Update**.
   Two updating methods are available.
   - **Rolling update**: Update pods one by one, which allows you to update the service without interrupting the business.
   - **Quick update**: Directly close all current pods, and launch the same number of new pods.

# Redeployment

Redeployment is to redeploy the containers under a service and pull a new image.

1. Click **Services** in the left navigation bar of TKE console to enter the service list, and click **More** -> **Redeploy**.


2. Click **OK**.

# Deleting Service

1. Click **Services** in the left navigation bar of TKE console to enter the service list, and click **More** -> **Delete**.


2. Click **OK**.


> **Note**:
> Deleting the service will delete all the pods and public network load balancers under the service. Back up your data in advance.

# Setting Limit on Service Resources

Last updated：2018-07-05 15:55:16

Resource Limits on Container Services

## About "Request" and "Limit"

`request` : The minimum amount of resources used by the container, a criterion for resources allocation when you schedule the container. The container is scheduled to the node only when the amount of resource that can be assigned on the node >= the number of container resources requests. Parameter Request does not limit the maximum available resources for the container.

`limit` : The maximum amount of resources used by the container. A value of 0 indicates no upper limit to the amount of resources.

> For more information on `limit` and `request` , click here

## CPU Limits

You can set Request and Limit for the amount of CPU, which is measured in cpu units (U) and expressed with decimal points.

> 1. CPU Request is used as a criterion for users to assign CPU resources for the container on the node during creation. It is called the Assigned CPU Resources.
> 2. CPU limit specifies the maximum amount of CPU to reserve for a container. A value of 0 indicates no upper limit to the amount of CPU (CPU Limit >= CPU Request).

## Memory Limits

You can only set the maximum amount of memory available for the container. Memory is measured in MiB, and expressed with decimal points.

> 1. Memory can't be scaled. When memory used by the container exceeds Memory Request, the container may be killed. Therefore, to ensure normal operation of the container, Request should be equal to Limit.
>    i. Memory Request (=Limit) is used as a criterion for users to assign memory resources for the container on the node during creation. It is called the Assigned Memory Resources.

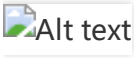
## CPU Usage Vs. CPU Utilization

> 1. CPU usage indicates the number of physical CPUs used by the container. CPU usage is used to determine whether CPU resource exceeds the Request and the Limit.

> 2. CPU utilization is the ratio of CPU usage to number of CPU single cores (or number of CPU cores on the node)

## Example

A simple example is given here to illustrate the role of Request and Limit. A 4U4G node is provided in the trial cluster. Two Pods (1, 2) are deployed, each with resources configuration of (CPU Request, CPU Limit, Memory Request, Memory Limit) = (1, 2U, 1G, 1G).
(1.0 GB = 1000 MiB)
The usage of CPU and memory on the node is shown as below:




Assigned CPU resources: 1U (to Pod 1) + 1U (to Pod 2) = 2U; unassigned CPU resources: 2U
Assigned memory resources: 1G (to Pod 1) + 1G (to Pod 2) = 2G; unassigned memory resources: 2G
In such case, one Pod with (CPU Request, Memory Request) = (2U, 2G), or two Pods with (CPU Request, Memory Request) = (1U, 2G) can be deployed on the node.

For Pod (1, 2), (CPU Limit, Memory Limit) = (2U, 1G). That is, the maximum amount of CPU available for a Pod is 2U when the resource is idle.

# Setting Access Mode of Services

Last updated：2018-07-05 15:56:07

The method for accessing a service determines the network attribute of this service. Different access methods offer different network capabilities. Tencent Cloud TKE provides four service access methods: Access from Public Network, Access Within Cluster Only, Access via Private Network in VPC and Not Enabled.

## Access from Public Network

For services accessed from public network, an entry (the public network load balancer) for Internet access is provided. The service accessed using this method can be accessed directly from the public network. This method is applicable to Web frontend services, such as wordpress frontend service.
For example, if we create a wordpress service that can be accessed from public network, the service access method is set to **Access From Public Network**. The created service can be accessed directly through **LB IP + Service Port**.

For more information on how to create a wordpress service, please see WordPress with Single Pod.

## Access Within Cluster Only

For services accessed within a cluster, an entry (service IP) of being accessed by other services or containers within the cluster is provided. This method is applicable to MySQL and other database services to ensure the isolation among service networks.
For example, if we create a MySQL service that can be accessed within a cluster only, the service access method is set to **Access Within Cluster Only**. The created service can be accessed directly through **Service IP/Name + Service Port**.

## Access via Private Network in VPC

For services accessed via the private network in a VPC, an entry (private network load balancer) of being accessed by other resources under the VPC in which the cluster resides is provided. This method is

applicable to services that need to be accessed by other clusters or CVMs under the same VPC.
For example, if we create a MySQL service that can be accessed via the private network in a VPC, the service access method is set to **Access via Private Network in VPC**. The created service can be accessed directly through **Private network LB IP + Service Port**.

## Not Enabled

If "Not Enabled" is selected for service access, no entry is provided for accessing from frontend service to the container. This can be used to discover or easily enable multiple container pods with custom service.

## More

In addition to the above four methods, you can also configure a layer-7 load balancer (HTTP/HTTPS) to forward to the service. For more information, please see Ingress Forwarding Configuration.
For more information on how to access within a cluster, please see Multiple Service Access Methods in Kubenerters and Configuration of Security Group in Tencent Cloud.