

# **Tencent Kubernetes Engine**

## **Load Balancers**

### **Product Documentation**



## Copyright Notice

©2013-2019 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

## Load Balancers

### Overview

### Forwarding Settings of Load Balancers

# Load Balancers

## Overview

Last updated : 2017-12-20 20:00:17

### Overview of Load Balancer

Load balancer of the container is the access entry to services created by the container. One load balancer is configured for one service frontend. The backend consists of multiple containers. Rather than the type of load balancer, you should focus on the features.

When the service is created:

- for public network, the TCP/UDP protocol-supported load balancer that can provide public network services is automatically created.
- for private network, the TCP/UDP protocol-supported load balancer that can provide private network services is automatically created.

If HTTP/HTTPS forwarding is required, you can create an HTTP/HTTPS load balancer on the load balancer page and configure the forwarding rules.

### Instructions

- [Basic Operations of Load Balancer](#)
- [Forwarding Configuration of Load Balancer](#)

# Forwarding Settings of Load Balancers

Last updated : 2019-05-28 09:55:06

## Prerequisites of using Ingress

Ingress service supports three scenarios as follows:

- Access from public network
- Access within cluster only
- Access via private network in VPC

Ingress supports application-based LB. An appropriate port needs to be enabled for the backend container node of an application-based LB. By default, CVM port is enabled for access from public network and access via private network in VPC, while CVM port is disabled for access within cluster only. However, if Ingress is set, CVM port is automatically enabled for the backend service. Services with access method disabled do not support setting of Ingress.

You can flexibly set an access method with Ingress for your service. The method for accessing a service does not conflict with Ingress.

## Wildcard in a domain name

A domain name must comply with the public network application-type load balancer domain name rules and Ingress domain name rules of Kubernetes:

1. It supports regular expression with a length limited to 1-80 characters.
2. Character sets supported by a non-regular domain name: a-z, 0-9, . and -.

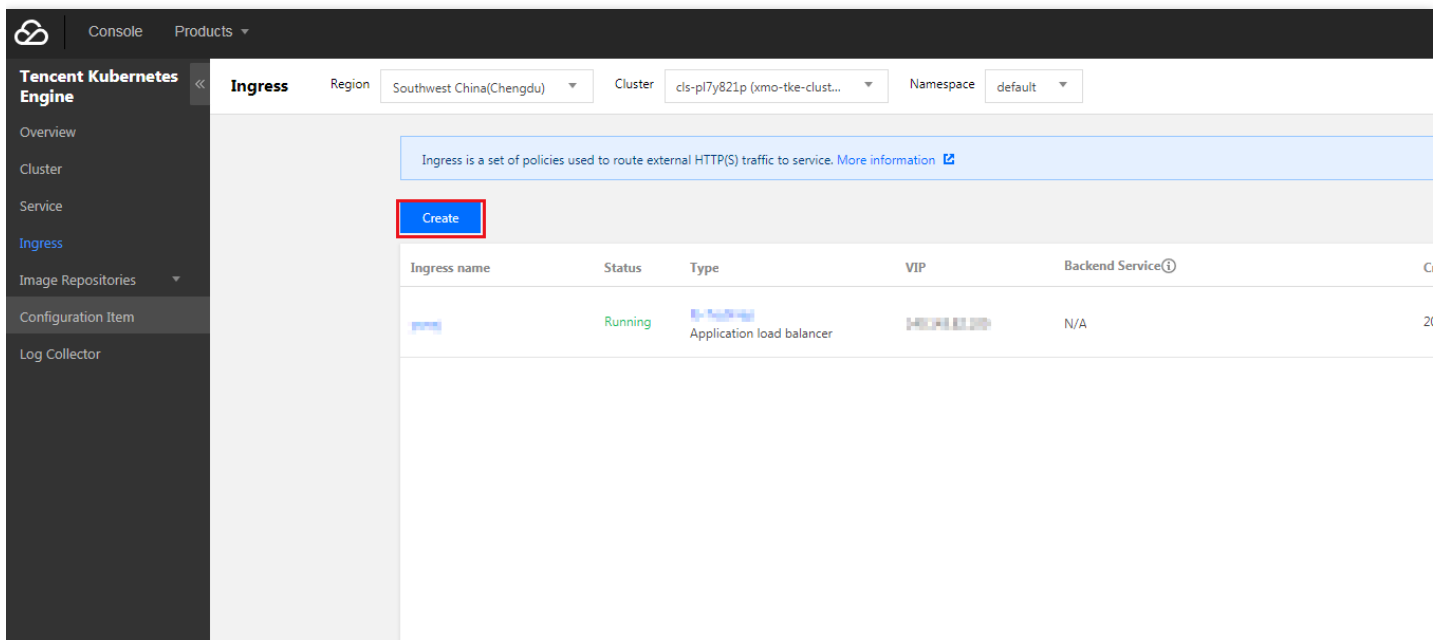
A domain name with wildcard only supports the format of `*.example.com`, and only one "\*" can be used in a domain name.

## Example of Ingress setting

Create a backend service that needs to use Ingress:

- hello service: Port 80 is listened with the entry file in `/path_hello/index.html`
- bye service: Port 80 is listened with the entry file in `/path_bye/index.html`

Create an Ingress on the Ingress page (skip this step if an Ingress already exists).



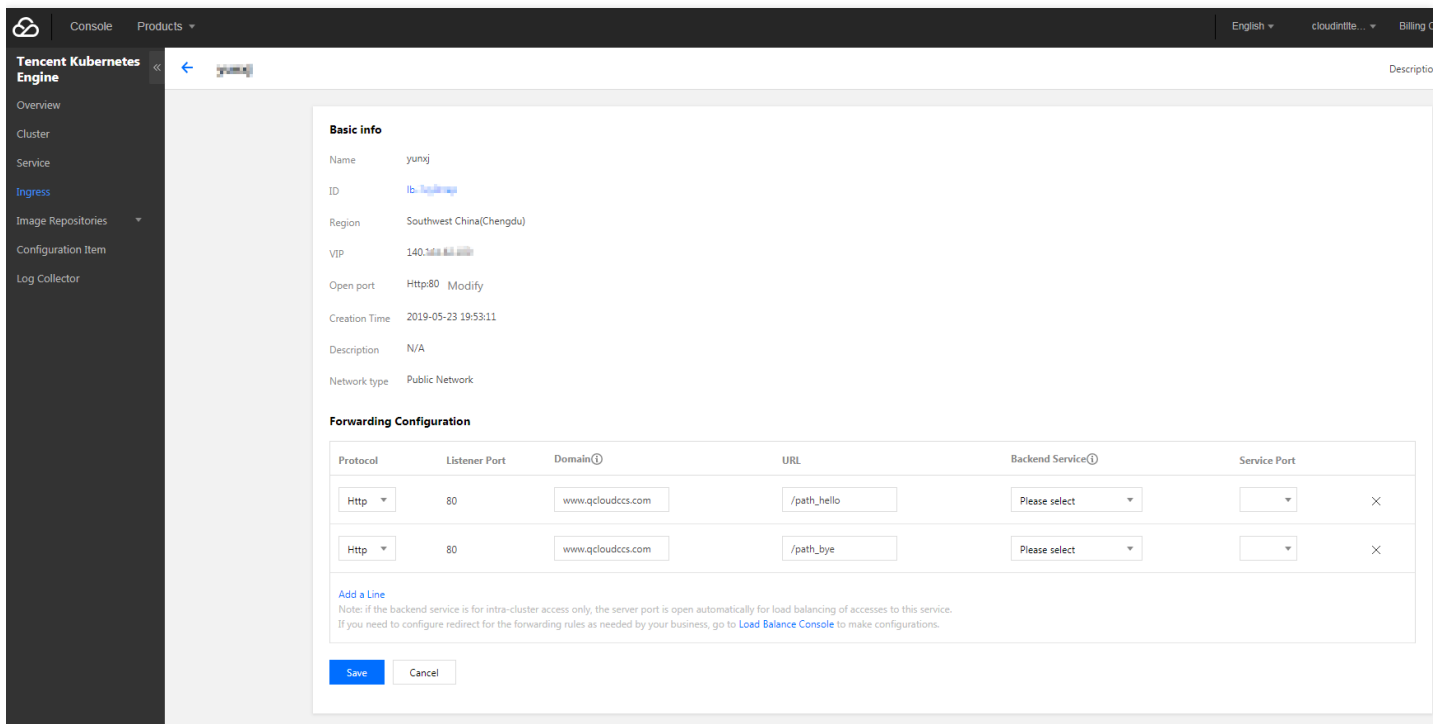
The screenshot shows the Tencent Kubernetes Engine console. On the left is a sidebar with navigation options: Overview, Cluster, Service, Ingress (selected), Image Repositories, Configuration Item, and Log Collector. The main area is titled 'Ingress' and includes filters for Region (Southwest China(Chengdu)), Cluster (cls-pl7y821p (xmo-tke-clust...)), and Namespace (default). A blue banner at the top states: 'Ingress is a set of policies used to route external HTTP(S) traffic to service. [More information](#)'. Below this, a red box highlights a 'Create' button. A table lists existing Ingress resources:

Ingress name	Status	Type	VIP	Backend Service①	
yunj	Running	Application load balancer	140.140.140.140	N/A	20

Resolve your domain name to the VIP of the load balancer. For more information, please see [Domain Name Resolution Help Documentation](#).

In the example below, [www.qcloudccs.com](http://www.qcloudccs.com) is resolved to sample load balancer.

Set Ingress forwarding rules:



The screenshot shows the 'Basic info' and 'Forwarding Configuration' sections of the Ingress resource 'yunxj'. The 'Basic info' section includes fields for Name, ID, Region, VIP, Open port, Creation Time, Description, and Network type. The 'Forwarding Configuration' section is a table with columns: Protocol, Listener Port, Domain①, URL, Backend Service①, and Service Port. It contains two rows of configuration for HTTP traffic on port 80, both pointing to 'www.qcloudccs.com' with paths '/path\_hello' and '/path\_bye'. Below the table, there is a note and a link to 'Add a Line'. At the bottom are 'Save' and 'Cancel' buttons.

Protocol	Listener Port	Domain①	URL	Backend Service①	Service Port	
Http	80	www.qcloudccs.com	/path_hello	Please select		×
Http	80	www.qcloudccs.com	/path_bye	Please select		×

[Add a Line](#)  
Note: if the backend service is for intra-cluster access only, the server port is open automatically for load balancing of accesses to this service. If you need to configure redirect for the forwarding rules as needed by your business, go to [Load Balance Console](#) to make configurations.

[Save](#) [Cancel](#)

Access test:

