

API Gateway Best Practice 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

 Tencent Cloud

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

Best Practice

API Gateway Providing the Access Capability for TKE

Integrating WAF to the API Gateway for Security Protection

Using API Gateway Dedicated Instance to Access Resources in IDC

Best Practice

API Gateway Providing the Access Capability for TKE

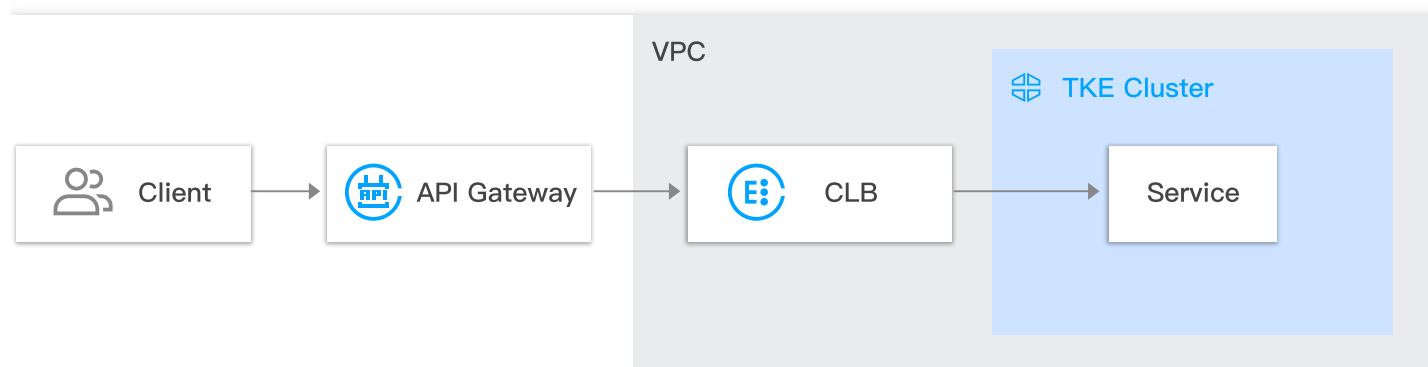
Last updated : 2020-12-01 11:21:30

Introduction to TKE

Based on native Kubernetes, Tencent Kubernetes Engine (TKE) is a container-oriented, highly scalable, and high-performance container management service. Compared with a client's container service, TKE has core advantages such as its ease of use, flexible expansion, security, reliability, high efficiency, and low costs. For more information, see [Tencent Kubernetes Engine](#).

Overview

As an open source platform for automated container operations, Kubernetes is a mainstream choice for developers. However, the access capability of Kubernetes clusters are not sufficient and cannot meet the requirements of large applications. Using API Gateway as the access layer of Kubernetes can significantly improve the access capability of Kubernetes clusters and empower Kubernetes clusters with advanced capabilities of API Gateway, adapting to more scenarios of more customers.



Prerequisites

You have activated Tencent Cloud services such as API Gateway, TKE, Cloud Load Balancer (CLB), Virtual Private Cloud (VPC), and Cloud Virtual Machine (CVM) and have permission to configure these

services, as they will be used during the configuration process.

Directions

Step 1: Create a VPC

1. Log in to the [VPC console](#).
2. In the left sidebar, click **Virtual Private Cloud** to access the VPC list page.
3. Click **+ New**. In the pop-up dialog box, set the parameters to create a VPC.
For more information, see [Managing VPC Instances](#).

Step 2: Create a CVM

1. Log in to the [CVM console](#).
2. In the left sidebar, click **Instances** to access the CVM instance list page.
3. Click **Create** to access the CVM purchase page.
4. Create a CVM by following the instructions in [Creating Instances via the CVM Purchase Page](#).

Note :

When creating a CVM, select the VPC and the subnet created in [Step 1](#) and retain the default values for the other parameters. In this example, a standard S5 CVM instance is created.

Step 3: Create a TKE cluster in the same VPC

1. Log in to the [TKE console](#).
2. In the left sidebar, click **Cluster** to access the TKE cluster list page.
3. Click **Create** at the top of the TKE cluster list. On the **Create Cluster** page, create a TKE cluster by following the instructions in [Creating a Cluster](#).

Note :

- When configuring the cluster information, set **Cluster network** to the VPC created in Step 1.
- When selecting a model, set **Node Source** to **Existing nodes** and **Master Node** to **Managed**, and select the CVM created in [Step 2](#) in the **Worker Configurations** area.
- Retain the default values for the other parameters.

Node Source Add Node Existing nodes

Master Node Managed Self-deployed

The default cluster's Master, Etcd and other components are maintained and managed by Tencent Cloud. For the convenience of management, you can also purchase CVM to deploy the Master. For details, please refer to ["Cluster Hosting Mode Instruction"](#)

Worker Configurations

Total CVMs: 4 0 selected

Enter the node name or full ID

<input type="checkbox"/>	ins-r6yqw86 as-tke-np-i45rud4g	Public IP: 119.29.35.92 Private IP: 10.0.8.12	i
<input type="checkbox"/>	ins-hp3j6z0g as-tke-np-i45rud4g	Public IP: 123.207.2... Private IP: 10.0.8.8	i
<input type="checkbox"/>	ins-ma4thhuo as-tke-np-i45rud4g	Public IP: 119.29.34... Private IP: 10.0.8.7	i
<input type="checkbox"/>	ins-n6el7d40 tke_cls-mi0xzjfi_worker	Public IP: 119.29.19... Private IP: 10.0.6.24	i

Press and hold Shift key to select more

Step 4: Create a nginx service in the TKE cluster

1. Log in to the [TKE console](#). In the left sidebar, click **Cluster** to access the TKE cluster list page.
2. Click the ID of the TKE cluster created in [Step 3](#) to access the cluster details page.
3. In the left sidebar, click **Workload** -> **Deployment** to access the deployment list page.
4. Click **Create** at the top of the deployment list page to access the workload creation page.
5. Set the parameters on the workload creation page by following the instructions in [Creating a Simple Nginx Service](#).

Note :

- Select a DockerHub nginx image.
- Set **Service Access** to **Via VPC**.
- Set **Load Balancer** to **Automatic creation**.
- In the **Port Mapping** area, set **Protocol** to **TCP**, and set both **Target Port** and **Port** to 80.

6. Click **Create Workload** to finish creating the Workload. TKE will automatically create the corresponding deployment and service.

Step 5: Create an API Gateway service

1. Log in to the [API Gateway console](#).

2. In the left sidebar, click **Service** to access the service list page.
3. Click **Create** at the top of the service list. In the pop-up dialog box, create an API Gateway service by following the instructions in [Creating Services](#).

Step 6: Create an API in the API Gateway service

1. In the [API Gateway console](#), click **Service** in the left sidebar to access the service list page.
2. Click the name of the created API Gateway service to access the service details page.
3. Click the **Manage API** tab, and then click **Create** to access the API creation page.
4. Enter the frontend configuration, the backend configuration, and the response result, and click **Complete** to finish creating the API.

Note :

When entering the backend configuration, set **Backend Type** to **HTTP**, **VPC Info** to the VPC created in Step 1, **VPC resources** to **CLB**, and **Backend Path** to **/**.

Step 7: Open the private IP ranges of API Gateway to the internet

1. Log in to the [CVM console](#). In the left sidebar, click **Security Groups** to access the security group list page.
2. Select a region and click **+ New**. In the pop-up dialog box, set the parameters and click **OK** to create a security group.
3. In the security group list, click the name of the created security group to access the security group details page. Click the **Security Group Rule** tab and then the **Inbound rule** tab to access the inbound rule list.
4. Click **Add a Rule**. In the pop-up dialog box, enter the following 5 private IP ranges of API Gateway: **9.0.0.0/8**, **10.0.0.0/8**, **100.64.0.0/10**, **11.0.0.0/8**, and **30.0.0.0/8**. Set **Protocol port** to **ALL** and **Policy** to **Allow** for the 5 private IP ranges, and click **Completed** to add the 5 inbound rules.
5. Return to the security group details page. Click the **Associate with Instance** tab and then the **Cloud Virtual Machine** tab. Click **Add Instances**. In the pop-up dialog box, associate the created security group with the CVM created in [Step 2](#) to open the private IP ranges of API Gateway to the internet.

Step 8: Publish and test the API Gateway service

1. In the [API Gateway console](#), click **Service** in the left sidebar to access the service list page.
2. Click the name of the created API Gateway service to access the service details page.
3. Click the **Basic Configuration** tab, and click **Publish** in the upper right corner of the page to publish the service to the **Publish** environment.
4. Request the API created in [Step 6](#). If the nginx page is displayed, the access is successful.

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Integrating WAF to the API Gateway for Security Protection

Last updated : 2020-10-09 11:51:27

Overview

Tencent Cloud Web Application Firewall (WAF) is an AI-based one-stop web service protection solution.

This document describes how to integrate WAF to the API Gateway to protect your APIs.

Prerequisites

- You have activated [Web Application Firewall](#).
- You have published APIs using the API Gateway.

Directions

Step 1. Bind a custom domain name in the API Gateway console

For more information about how to bind a custom domain name in the API Gateway console, refer to [Custom Domain Name and Certificate](#).

Custom Domain Name Binding Guide

1

Get Domain Name

Go to [Domain Name Registration](#) or get a domain name from a domain name registrar

2

Tencent Cloud ICP Filing Registration

For the processes of ICP filing in Tencent Cloud, you can refer to [ICP Filing Registration](#)

3

Configure CNAME and Resolve to Second-Level Domain

Add a CNAME record and resolve the domain name to the second-level domain name^①

4

Bind to Take Effect

Bind a custom domain name, which will take effect immediately after configuration

[Create](#) ↻

Domain Name	Path Mapping	Protocol	Network Type	SSL Certificate	Operation
No data yet					

Total items: 0 20 / page ⏪ ⏩ 1 / 1 page ⏪ ⏩

Note :

When a custom domain name is bound to the API Gateway console, the system will check whether you have configured CNAME and resolved it to the service subdomain name. Be sure to first configure CNAME and resolve it to the subdomain name of the API Gateway, modify the CNAME record, and point the custom domain name to the WAF domain name.

Step 2. Configure WAF

1. Log in to the [WAF console](#).
2. Click **Web Application Firewall** -> **Defense settings** in the left sidebar to access the domain name list page.
3. Click **Add domains** at the top of the **Domain Name List** module.
4. Enter the **Real Server Address** and the subdomain name of the API Gateway, and complete the other configurations.

Domain Configuration

Domain Name

Web server configurations ⓘ HTTP 80 [Other ports](#)
 HTTPS

Enable HTTP2.0 ⓘ No Yes
Please make sure your real server supports and enables HTTP2.0. Otherwise it will be degraded to HTTP1.1

Real Server Address ⓘ IP Domain Name

Please enter the real server domain name. It cannot be the same as the protection domain name

Other Configuration

Proxy No Yes
Choose Yes if you are using proxies (Dayu, CDN or acceleration service)

Enable WebSocket No Yes
If you website uses WebSocket, please select "Yes"

Load Balance Round-Robin IP Hash

5. Click **Save**. The domain name should now be in the “CNAME record not configured” status.

Domain Name List							
Domain Name	Protection st...	VIP ⓘ	Usage Mode	Intermediate IP ⓘ	Access Log S...	WAF Sw...	Operation
<input type="checkbox"/> www.apigw.tencentcs.com	Parsing failed ⓘ	111.231.254.138	Rule: Blocking mode	111.230.122.0/24 total 11 View	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Delete Edit Defense configuration

Step 3. Modify the CNAME record

1. Modify the CNAME record and resolve the custom domain name to the WAF domain name.
2. Log in to WAF console and click **Web Application Firewall** -> **Defense settings** in the left sidebar to access the domain name list page.
3. Click the refresh icon in the **Protection status** column. The status should change to **Normal protection**.

Using API Gateway Dedicated Instance to Access Resources in IDC

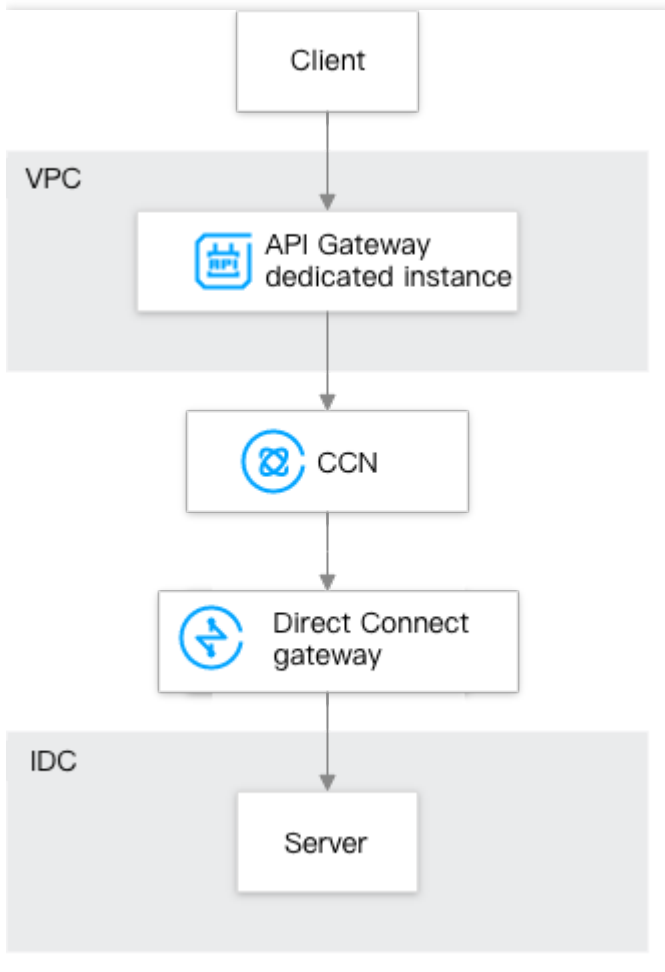
Last updated : 2021-08-13 11:24:13

Overview

When you use a hybrid cloud architecture, your business may be deployed in both public and private clouds, but a unified access layer is required, which serves as the ingress and egress of traffic and supports the processing of all non-business features such as authentication and traffic throttling.

An API Gateway dedicated instance runs in a VPC and supports forwarding client requests to various services deployed in the VPC or local IDC or on the public network. It is deeply integrated with common backend services to provide a productized connection method. Therefore, it is very suitable as a unified access layer in complex network environments. This document describes how to connect to backend resources in an IDC by using an API Gateway dedicated instance.

Scheme Advantages



- The API Gateway dedicated instance can forward requests to the backend resources deployed in the VPC and local IDC and on the public network at the same time, seamlessly connecting the cloud and local systems and enabling smooth cloudification.
- The rich features provided by API Gateway can also be used, such as IP access control, traffic throttling, and log monitoring.
- Resources on the private network can be interconnected with each other through CCN, fine-grained routing is supported to guarantee the quality, and diversified tiered pricing is supported to reduce the costs.

Directions

Step 1. Create a CCN instance and associate it with a network instance

1. Log in to the [VPC console](#).
2. On the left sidebar, click **Cloud Connect Network** to enter the CCN management page.
3. Click + **New**.

4. In the pop-up window, enter the name and description of the CCN instance and select the billing mode, service level, and bandwidth limit mode.
5. Associate the IDC's Direct Connect gateway with the VPC.

Create CCN instance ✕

Name

Billing Mode ⓘ Pay-as-you-go by monthly 95th percentile
The default bandwidth cap is 1 Gbps. It's billed based on the actual bandwidth usage of the current month on a [95th percentile basis](#)

Service Level ⓘ Platinum ⓘ Gold ⓘ Silver ⓘ

Bandwidth limit mode ⓘ Regional Outbound Bandwidth Cap Inter-region bandwidth cap

Description

Associated Instances

VPC	Please select	Search for VPC name or ID	✕
VPN Gateway	Please select	Search by VPN gateway nam	✕

[Add](#)

[Advanced Options](#) ▼

Step 2. Purchase an API Gateway dedicated instance

1. Log in to the [API Gateway console](#) and select **Instance** on the left sidebar.
2. Click **Create** to enter the API Gateway dedicated instance purchase page.
3. Select and enter the instance configuration information.

Note :

The VPC configuration of the dedicated instance should be the same as that of the VPC instance associated with the CCN instance created in [step 1](#).

4. Click **Buy Now** and make the payment.

Step 3. Create a service and API under the instance

1. Log in to the [API Gateway console](#) and select **Service** on the left sidebar.
2. Click **Create** and select **Dedicated** as the instance type.
3. In the pop-up window for instance selection, select the dedicated instance purchased in step 1 and click **Submit**.
4. Click the name of the created service in the service list to enter its API management page.
5. Click **Create** to enter the API creation page and set the configuration items. Select public URL/IP as the API backend type, enter the IDC's private IP, port, and path to be bound as the backend address, and click **Submit**.
6. Request the created API, and you will see that the backend resources in the IDC can be accessed through API Gateway.