Cloud Load Balancer

Product Introduction

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

Product Introduction
   Introduction
   Strengths
   Use Cases
   Principles
   Restrictions
Application and Conventional CLB comparison
What is Cloud Load Balancer?

Cloud Load Balancer is a traffic delivery service for multiple CVMs. Cloud load balancer can extend application systems' external service capabilities through traffic delivery and improve their availability by eliminating single point of failure.

The Cloud Load Balance service virtualizes multiple CVM resources located in the same region into a high-performance, high-availability application service pool by setting a virtual service address (VIP), and delivers the network requests from the client to the CVM pool in a way specified by the application.

Cloud Load Balance can check the health of CVMs in the CVM Pool, and automatically isolate abnormal CVMs, thus dealing with the single point of failure (SPOF) of a single CVM and improving the overall service capability of applications.

Tencent Cloud Load Balance provides self-management, self-troubleshooting, anti-network attacks and other advanced features for enterprises, communities, e-commerce, games and other user scenarios.

Components

A cloud load balancer group that provides services typically consists of the following components:

- CloudLoadBalancer: A cloud load balancer instance for traffic delivery
- VIP (virtual IP): An IP address through which the cloud load balancer provides service to the client
- Backend/Virtual Server: A group of cloud load balancer instances on the backend for processing requests
- VPC/Basic network: The overall network environment

Access requests from other servers other than cloud load balancer is delivered through the cloud load balancer instance to the backend CVM for processing according to the associated policies and forwarding rules.
Glossary

<table>
<thead>
<tr>
<th>Term</th>
<th>Full Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud Load balancer</td>
<td>Cloud Load Balancer</td>
<td>A network cloud load balance service provided by Tencent Cloud that can be combined with CVM to offer cloud load balance service based on TCP/UDP and HTTP</td>
</tr>
<tr>
<td>Load Balance Listener</td>
<td>Load Balance Listener</td>
<td>Including listening port, load balance policy and health check configuration, each item to be listened corresponds to a back-end application service</td>
</tr>
<tr>
<td>Back-end Server</td>
<td>Real Server</td>
<td>A group of CVM instances that accept load balance delivery requests. The cloud load balancing service forwards access requests to this group of back-end CVMs according to user-defined rules</td>
</tr>
<tr>
<td>VIP</td>
<td>Virtual IP</td>
<td>The service address assigned by the system, which currently is an IP address. You can choose whether to open it to the internet so as to create a private-network CLB instance or a public-network CLB instance</td>
</tr>
</tbody>
</table>

How Does Cloud Load Balancer Work?

Working Principle

The cloud load balancer accepts incoming traffic from the client and routes the request to a back-end CVM instance in one or more available zones for processing.

Cloud Load Balance service is mainly provided by cloud load balancer listener. The listener is responsible for monitoring requests on the cloud load balancer instance, delivering policies to back-end CVMs and other services. By configuring the forwarding protocols and protocol ports of "client-cloud load balancer" and "cloud load balancer-backend CVM", cloud load balancer can forward the request directly to the back-end CVM.

It is recommended that you configure a back-end CVM instance of the cloud load balancer across multiple available zones. If an available area becomes unavailable, the cloud load balancer routes the traffic to other available zones for normal operation, thereby shielding off the service interruption caused by a single available zone failure.

Request Routing Selection
The client requests to access the service through domain name. Before the request is sent to the cloud load balancer, the DNS server will resolve the load balancing domain name and return the requested CVM IP address to the client. When the cloud load listener is requested, a different cloud load balancing algorithm will be used to deliver the request to the back-end CVM. Tencent Cloud currently supports Weighted Round-Robin and ip_hash balancing algorithms. It will support the Weighted Least-Connection Scheduling in the future.

**Monitoring Back-end Service Status**

The cloud load balancer also monitors the running status of back-end instances, so as to ensure that traffic is only routed to the normal running instance. When the cloud load balancer detects an abnormal instance, it stops routing traffic to that instance until the instance is detected to return to normal again.

**Related Services**

Cloud Load Balance can be used with the following services to improve application availability and scalability:

- **CVM Instance**: enables the application to run on the cloud virtual server. For more information, refer to [CVM Product Documentation](#).
- **Auto Scaling**: Controls the number of instances flexibly. When a cloud load balancer instance is enabled in Auto Scaling, the scaled instance is automatically added to the cloud load balancing group, and the terminated instance is automatically moved out of the cloud load balancing group. For more information, see [Auto Scaling Product Documentation](#).
- **Cloud Monitor**: Helps you monitor the running status of cloud load balancer and all back-end instances and perform the operations accordingly. For more information, refer to [Cloud Monitor Product Documentation](#).
Strengths

Last updated: 2020-03-09 14:59:21

CLB performance is evaluated mainly based on the following metrics:

- **TPS (connection per second):** the number of TCP connections created by a CLB instance per second.
- **Maximum number of concurrent connections:** the total number of established TCP connections when client sends requests to the server, i.e., the total number of TCP connections created by the server per second.
- **QPS (query per second):** also known as request per second (RPS). It is the number of GET, POST, and HEAD packets sent by the client to the HTTP service after a connection is established.
- **Throughput:** the total traffic/bandwidth supported by a CLB instance.

Tencent Cloud provides high-performance CLB services:

- A single CLB cluster can provide over 120 million concurrent connections and respond to hundreds of millions of web access requests.
- A single CLB cluster can handle a peak traffic of 40 GB/s with up to 6 million PPS (packets per second).
- CLB strictly isolates the traffic of each tenant and provides active protection against DDoS attacks. If your service is under DDoS attack, CLB will provide you free defense capability against 2–10 GB of attack traffic.

If you have higher protection requirements, you can purchase Anti-DDoS Pro, which helps defend against up to 300 GB of attack traffic.

To protect the application layer, you can purchase Tencent Cloud Web Application Firewall (WAF). It protects web security at the application layer against web vulnerability attacks, malicious crawlers, and CC attacks.
Use Cases

Last updated: 2020-03-09 15:04:07

CLB is mainly suitable for the following scenarios:

- Traffic distribution. CLB can distribute the traffic of business with a large number of access requests to multiple CVM instances.
- Elimination of single point of failure. When some CVM instances become unavailable, CLB can automatically block them to ensure the normal operation of the application system.
- Horizontal scalability. You can scale out the service capability of application systems as needed, which is suitable for web and app servers.
- Global load balancing. With Tencent Cloud DNS, CLB can support global and multi-regional load balancing for remote disaster recovery.

Traffic distribution and elimination of single point of failure

You can use CLB to distribute business traffic to multiple CVM instances.

- Business client accesses CLB.
- Multiple CVM instances form a high-performance and high-availability service pool, to which CLB forwards the business traffic.
- When one or more CVM instances become unavailable, CLB can automatically block them and distribute the requests to normal CVM instances, ensuring the operation of the application system.
- The session persistence feature can forward requests from the same client to the same real server, improving access efficiency.

Horizontal Scalability

With Auto Scaling (AS), CLB can automatically create and release CVM instances based on your business needs.

- You can configure auto scaling policies to manage the number of CVM instances, deploy the instance environment, and ensure the operation of your business. CLB can automatically add CVM instances when demands peak to keep high performance, while removing CVM instances when demands drop to reduce costs.
- For example, during major sales campaigns in ecommerce such as Black Friday, web traffic may suddenly increase by 10 times and lasts only for a few hours. In this case, CLB and AS can be used...
to maximize IT cost savings.

Global Load Balancing

With Tencent Cloud DNS, you can resolve your business traffic to global and multi-regional CLB instances to implement active-active and remote disaster recovery across regions.

- You can deploy CLB instances in different regions and bind them to CVM instances in corresponding regions.
- You can use Tencent Cloud DNS to resolve domain names to the CLB VIP in each region.
- Business traffic will be forwarded to multiple CVM instances in multiple regions via DNS and CLB, achieving global load balancing.
- When a region becomes unavailable, you can suspend resolution of the CLB VIP in that region to ensure that your business is not affected.
Principles
Last updated: 2020-03-10 10:39:36

CLB provides Layer-4 (TCP, UDP, and TCP SSL protocols) and Layer-7 (HTTP and HTTPS protocols) load balancing. You can use CLB to distribute business traffic to multiple real servers to eliminate single point of failure and guarantee business availability. CLB adopts cluster deployment to achieve session synchronization, eliminating server’s single point of failure and improving system redundancy to ensure service stability. CLB can be deployed in multiple data centers in the same region to implement intra-city disaster recovery.

Infrastructure

Currently, Tencent Cloud CLB provides Layer-4 and Layer-7 load balancing services:

- At Layer-4, load balancing is implemented based on the unified Tencent Gateway (TGW). TGW has features such as high availability, high scalability, high performance, and strong anti-attack capability. It supports high-performance forwarding based on Data Plane Development Kit (DPDK). With TGW, a single cluster can support hundreds of millions of concurrent requests and tens of millions of packets per second (PPS). Many Tencent businesses, such as Tencent Games, Tencent Video, WeChat, and QQ, use TGW for service access.

- At Layer-7, load balancing is implemented based on Secure Tencent Gateway (STGW). It is a load balancing service developed by Tencent based on Nginx that supports large-scale concurrence. It carries a large amount of Tencent’s Layer-7 business traffic, such as Tencent News, Licaitong,
Forwarding Path

CLB forwards business traffic and real servers process business requests. CLB communicates with real servers via Tencent Cloud private network. Both TGW and STGW are deployed on multiple servers, and provide load balancing services through clusters. The forwarding path of CLB is as
1. **TCP and UDP protocol:**
   - The forwarding logic of TCP/UDP protocol is processed by TGW cluster.
   - After receiving the business traffic, TGW forwards it to real servers via Tencent Cloud's private network. The return packets from real servers are also returned to the client via TGW.

2. **TCP SSL protocol**
   - When TCP SSL protocol is processed, business traffic passes through the TGW cluster and then STGW cluster, which forwards the traffic to real servers.
   - Before a new session is established, it must pass through the accelerator card cluster for certificate verification, encryption, decryption and other operations.
   - When business traffic arrives, it passes through TGW, STGW, and real servers in sequence via Tencent Cloud's private network. The return packets are sent to the client in reverse sequence.

3. **HTTP and HTTPS protocols**
   - When HTTP or HTTPS protocol is processed, business traffic passes through the TGW cluster and then STGW cluster, which identifies the HTTP protocol and forwards the traffic to real servers.
   - Before a new HTTPS session is established, it must pass through the accelerator card cluster for certificate verification, encryption, decryption and other operations. HTTPS will be converted to HTTP protocol and then forwarded to real servers.
   - When business traffic arrives, it passes through TGW, STGW, and real servers in sequence via Tencent Cloud's private network. The return packets are sent to the client in reverse sequence.
The use of Tencent Cloud CLB has certain restrictions, and different types of CLB instances have their own use limits. For more information on CLB instance types, please see Instance Types.

<table>
<thead>
<tr>
<th>Instance Type</th>
<th>Resource</th>
<th>Default Restriction</th>
</tr>
</thead>
<tbody>
<tr>
<td>General restrictions for all instances</td>
<td>Number of public network instances can be created under one account in a single region</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Number of private network instances can be created under one account in a single region</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Number of listeners can be added to an instance</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>Ports that can be selected by a listener in an instance</td>
<td>An integer between 1 - 65535</td>
</tr>
<tr>
<td>CLB (formerly &quot;application CLB&quot;)</td>
<td>Number of domain name and URL forwarding rules can be configured for an HTTP/HTTPS listener in a CLB instance</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>Number of servers can be bound to a forwarding rule in a CLB instance</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Number of backend ports that can correspond to a frontend port of a CLB instance</td>
<td>Multiple ports</td>
</tr>
<tr>
<td>Classic CLB</td>
<td>Number of servers can be bound to a listener in a classic CLB instance</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Number of backend ports that can correspond to a frontend port of a classic CLB instance</td>
<td>1 port</td>
</tr>
</tbody>
</table>

A CLB instance **will not unbind itself** from the CVM instance. After a CVM instance becomes isolated (pay-as-you-go CVM instance has been in arrears for more than 2 hours), it **will not unbind itself** from the CLB instance either.
Application and Conventional CLB comparison

CLB offers two types of instances: CLB (formerly "application CLB") and classic CLB.

- Cloud Load Balancer: support TCP/UDP/HTTP/HTTPS Protocol, provide balance based on domain name and URL path, and support flexible repost.
- Traditional Cloud Load Balancer: private network does not support HTTP/HTTPS Protocol, and the configuration method is simple.

Cloud Load Balancer can cover all the functions of the traditional Cloud Load Balancer. From Features, product performance and other considerations, it is recommended that you use the instance type is Cloud Load Balancer. A detailed comparison of the two is as follows:

<table>
<thead>
<tr>
<th>Product Type</th>
<th>Cloud Load Balance</th>
<th>Classic Cloud Load Balancer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Public network</td>
<td>Private network</td>
</tr>
<tr>
<td>Layer-7 repost (HTTP/HTTPS)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Layer-4 repost (TCP / UDP)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Layer-4 encrypts repost (TCP SSL)</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>Support for HTTP/2 and websocket (secure)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Cloud Load Balancer's strategy</td>
<td>IP hash (Layer-7)</td>
<td>IP hash (Layer-7)</td>
</tr>
<tr>
<td></td>
<td>Weighted polling</td>
<td>Weighted polling</td>
</tr>
<tr>
<td></td>
<td>Weighted Connections</td>
<td>Weighted Connections</td>
</tr>
<tr>
<td></td>
<td>Minimum Connections</td>
<td>Minimum Connections</td>
</tr>
<tr>
<td>Feature</td>
<td>Service 1</td>
<td>Service 2</td>
</tr>
<tr>
<td>--------------------------------------------------------------</td>
<td>-----------</td>
<td>-----------</td>
</tr>
<tr>
<td>Session hold</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Health check</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Customize repost rules (domain name / URL)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Support SNI multi-certificate feature</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Repost to different back-end ports</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Layer-7 personalized configuration</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Layer-7 redirect function (rewrite)</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>Support cross-region binding function</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>Support Layer-7 log storage to COS</td>
<td>✓</td>
<td>×</td>
</tr>
</tbody>
</table>