

Alibaba Cloud

Server Load Balancer
Health check

Document Version: 20220406

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).
5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.
6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions









Style	Description	Example
 Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 Danger: Resetting will result in the loss of user configuration data.
 Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 Warning: Restarting will cause business interruption. About 10 minutes are required to restart an instance.
 Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: If the weight is set to 0, the server no longer receives new requests.
 Note	A note indicates supplemental instructions, best practices, tips, and other content.	 Note: You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click Settings> Network> Set network type .
Bold	Bold formatting is used for buttons, menus, page names, and other UI elements.	Click OK .
<code>Courier font</code>	Courier font is used for commands	Run the <code>cd /d C:/window</code> command to enter the Windows system folder.
<i>Italic</i>	Italic formatting is used for parameters and variables.	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] or [a b]	This format is used for an optional value, where only one item can be selected.	<code>ipconfig [-all -t]</code>
{ } or {a b}	This format is used for a required value, where only one item can be selected.	<code>switch {active stand}</code>

Table of Contents

1. Health check overview	05
2. Configure health checks	11
3. Execute a health check script	14
4. Disable the health check feature	16
5. Health check FAQ	17

1. Health check overview

Server Load Balancer (SLB) checks the service availability of backend servers (ECS instances) by performing health checks. Health checks improve the overall availability of your frontend service, and avoid service impacts caused by exceptions of backend ECS instances.

After you enable the health check function, SLB stops distributing requests to the instance that is discovered unhealthy and restarts forwarding requests to the instance only when it is declared healthy.

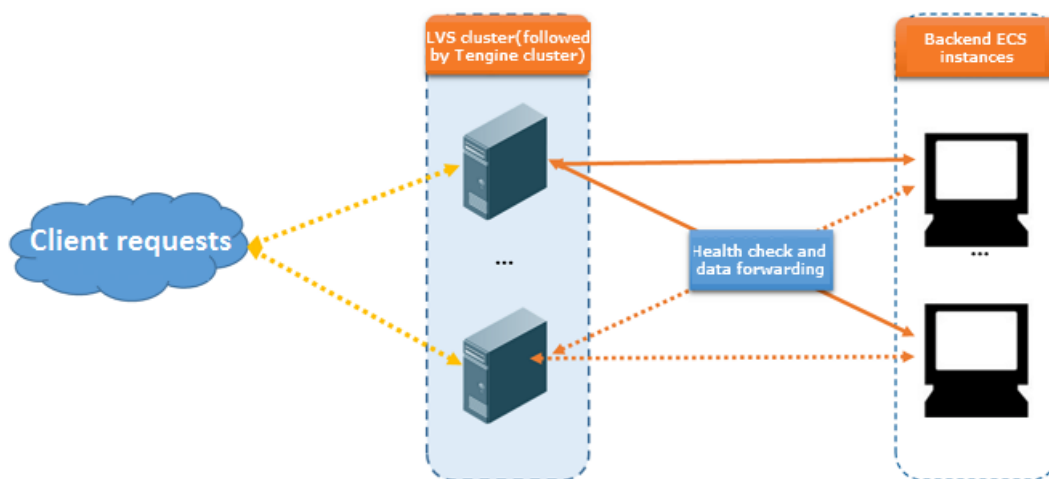
If your service is highly sensitive to traffic load, frequent health checks may impact your service. To reduce the impact on your service, you can reduce the health check frequency, increase the health check interval, or change a layer-7 health check to a layer-4 one based on the service conditions. To guarantee the service availability, we do not recommend disabling the health check function.

Health check process

SLB is deployed in clusters. Data forwarding and health checks are handled at the same time by node servers in the LVS cluster and Tengine cluster.

The node servers in the cluster independently perform health checks in parallel, according to health check configurations. If an LVS node server detects that a backend ECS instance fails, the LVS node server no longer sends new client requests to this ECS instance. This operation is synchronized among all node servers.

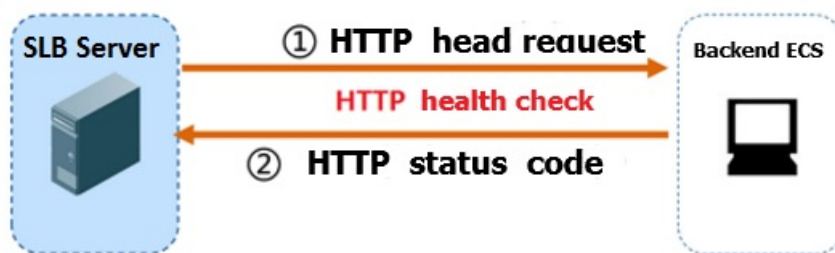
The IP address range used for health checks is 100.64.0.0/10. Make sure that backend ECS instances do not block this CIDR block. You do not need to configure a security group rule to allow access from this CIDR block. However, if you have configured security rules such as iptables, you must allow access from this CIDR block. (100.64.0.0/10 is reserved by Alibaba Cloud. Other users cannot use any IP address in this CIDR block and therefore no security risks exist.)



Health checks of HTTP/HTTPS listeners

For layer-7 (HTTP or HTTPS) listeners, SLB checks the status of backend servers by sending HTTP HEAD requests, as shown in the following figure.

For HTTPS listeners, certificates are managed in SLB. HTTPS is not used for data exchange (including health check data and service interaction data) between SLB and backend ECS instances so that the system performance is improved.

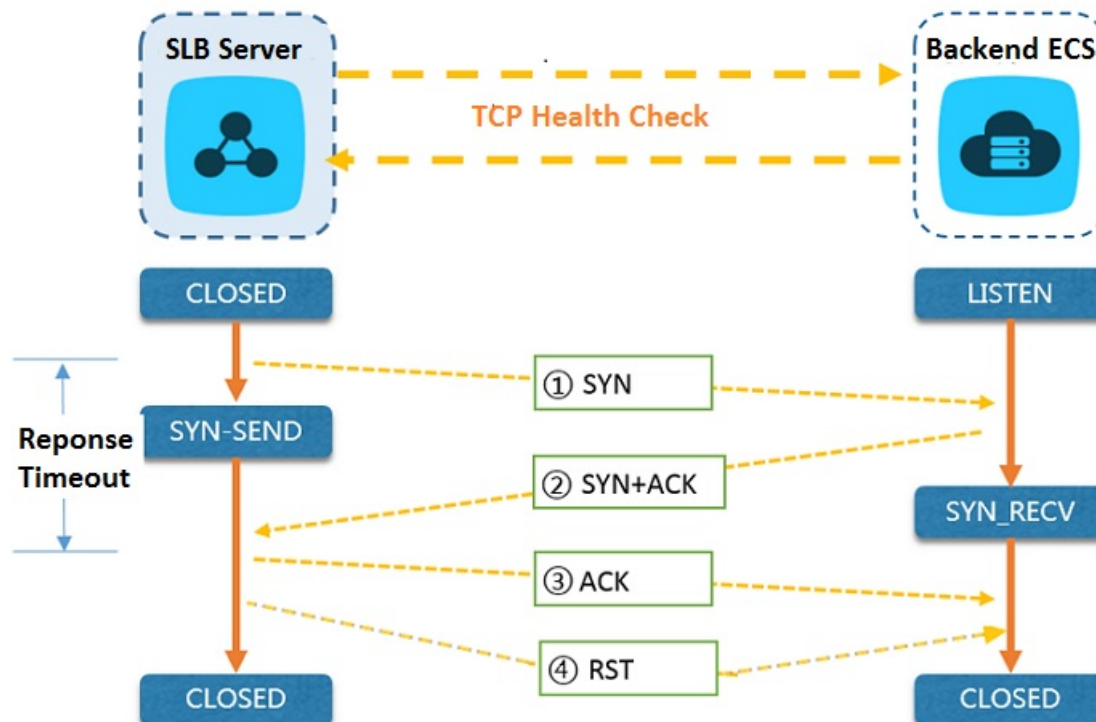


The health check process of a layer-7 listener is as follows:

1. A T Engine node server sends an HTTP HEAD request to the internal IP address, health check port, and health check path of a backend server according to the health check settings.
2. After receiving the request, the backend server returns an HTTP status code based on the running status.
3. If the T Engine node server does not receive the response from the backend server within the specified response timeout period, the backend server is declared as unhealthy.
4. If the T Engine node server receives a response from the backend ECS instance within the specified response timeout period, the node server compares the response with the configured status code. If the status code is the same, the backend server is declared as healthy. Otherwise, the backend server is declared as unhealthy.

Health checks of TCP listeners

For TCP listeners, SLB checks the status of backend servers by establishing TCP connections, as the following figure shows.



The health check process of a TCP listener is as follows:

1. The LVS node server sends a TCP SYN packet to the internal IP address and health check port of a backend ECS instance.

2. After receiving the request, the backend server returns a TCP SYN and ACK packet if the corresponding port is listening normally.
3. If the LVS node server does not receive the packet from the backend ECS instance within the specified response timeout period, the server determines that the service does not respond and health check fails. Then, the server sends an RST packet to the backend ECS instance to terminate the TCP connection.
4. If the LVS node server receives the packet from the backend ECS instance within the specified response timeout period, the server determines that the service runs properly and the health check succeeds. Then, the server sends an RST packet to the backend ECS instance to terminate the TCP connection.

Note In general, TCP three-way handshakes are conducted to establish a TCP connection. After the LVS node server receives the SYN and ACK data packet from the backend ECS instance, the LVS node server sends an ACK data packet, and then immediately sends an RST data packet to terminate the TCP connection.

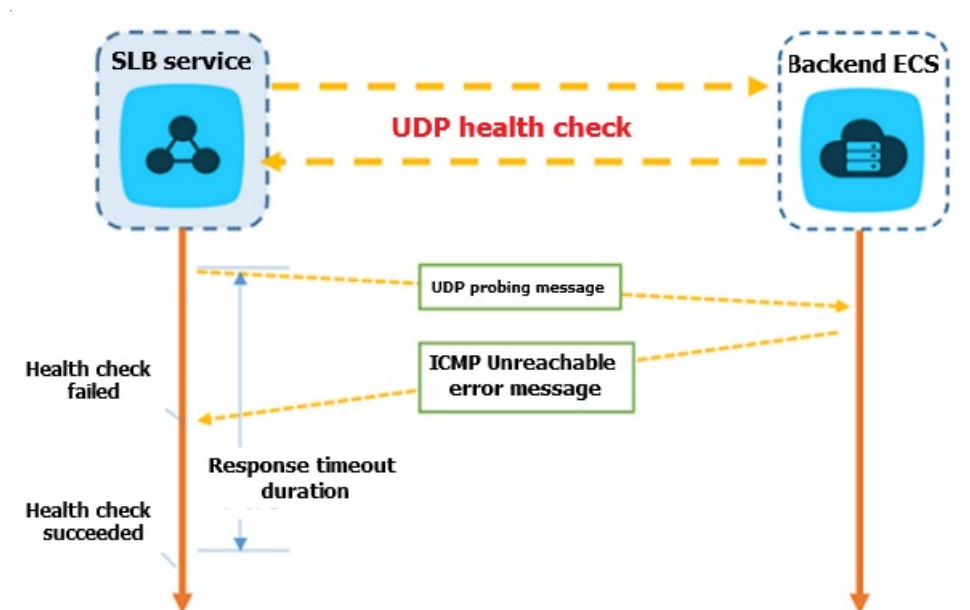
This process may cause backend server to think an error occurred in the TCP connection, such as an abnormal exit, and then report a corresponding error message, such as `Connection reset by peer`.

Solution:

- Use HTTP health checks.
- If you have enabled the function of obtaining real IP addresses, you can ignore the connection errors caused by accessing the preceding SLB CIDR block.

Health checks of UDP listeners

For UDP listeners, SLB checks the status of backend servers by sending UDP packets, as shown in the following figure.




The health check process of a UDP listener is as follows:

1. The LVS node server sends a UDP packet to the internal IP address and health check port of the ECS

instance according to health check configurations.

2. If the corresponding port of the ECS instance is not listening normally, the system returns an ICMP error message, such as `port XX unreachable`. Otherwise, no message is sent.
3. If the LVS node server receives the ICMP error message within the response timeout period, the ECS instance is declared as unhealthy.
4. If the LVS node server does not receive any message within the response timeout period, the ECS instance is declared as healthy.

 **Note** For UDP health checks, the health check result may fail to reflect the real status of a backend ECS instance in the following situation:

If the ECS instance uses a Linux operating system, the speed of sending ICMP messages in high traffic hours is limited due to the anti-ICMP attack protection function of Linux. In this case, even if an exception occurs to the ECS instance, SLB may declare the backend server as healthy because the error message `port XX unreachable` is not returned. Then, the health check result deviates from the actual service status.

Solution:

Specify a pair of request and response for UDP health checks. If the specified response is returned, the ECS instance is considered healthy. Otherwise, the ECS instance is considered unhealthy. To achieve this, you must configure the client accordingly.

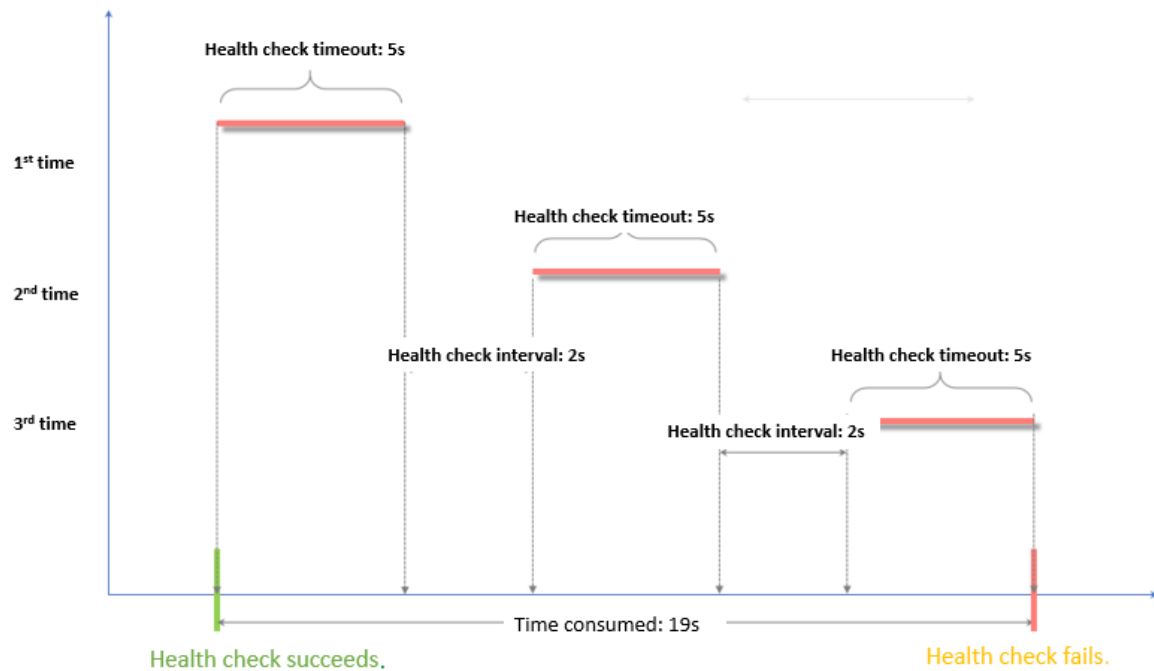
Health check time window

The health check function effectively improves the availability of your service. However, to avoid the impact of switching caused by frequent health check failures on system availability, status is switched (health check succeeded or failed) only when the health check succeeds or fails for a specified number of times in the time window. The health check time window is determined by the following three factors:

- Health check interval: how often the health check is performed.
- Response timeout: the length of time to wait for a response.
- Health check threshold: the number of consecutive successes or failures of health checks.

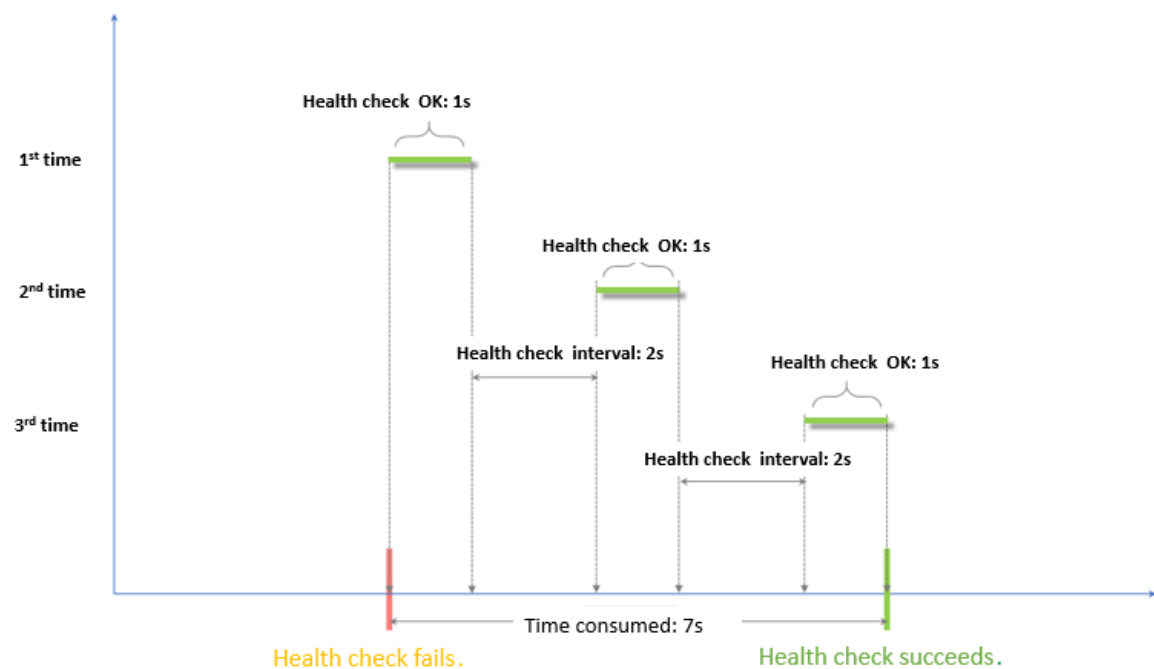
The health check time window is calculated as follows:

- Health check failure time window = Response timeout × Unhealthy threshold + Health check interval × (Unhealthy threshold - 1)



- Health check success time window = Response time of a successful health check × Healthy threshold + Health check interval × (Healthy threshold - 1)

Note The success response time of a health check is the duration from the time when the health check request is sent to the time when the response is received. When TCP health checks are used, the time is short and almost negligible because the health check only checks whether the port is alive. For HTTP health checks, the time depends on the performance and load of the application server and is generally within seconds.

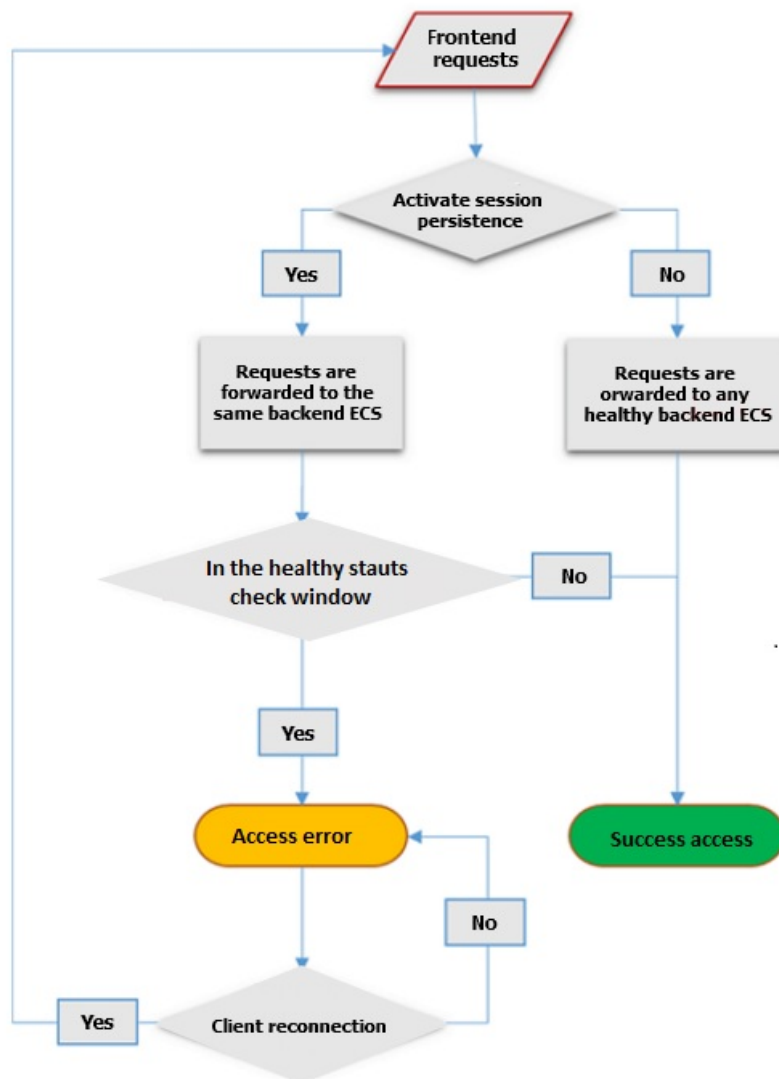


The health check result has the following impact on request forwarding:

- If the health check of the target ECS instance fails, new requests are distributed to other ECS

instance. The client access is normal.

- If the health check of the target ECS instance succeeds, new requests are distributed to it. The client access is normal.
- If a request arrives during a health check failure window, the request is still sent to the ECS instance because the ECS instance is being checked and has not been declared unhealthy. Then, the client access fails.



2.Configure health checks

This topic describes how to configure health checks. You can configure health checks when you create a listener or for an existing listener. The default health check settings can meet your requirements in most cases.

Context


You can configure health checks by using the SLB console or API operations. For more information, see [Health check overview](#) and [Health check FAQ](#).


Procedure

- 1.
- 2.
3. Select a region.
4. Find an SLB instance and click the instance ID.
5. On the page that appears, click the **Listener** tab.
6. Click **Add Listener**, or find an existing listener and click **Modify Listener** in the **Actions** column.
7. Click **Next** to go to the **Health Check** step and configure the health check.

We recommend that you use the default settings when you configure health checks.

Health check parameters

Parameter	Description
Health Check Protocol	<p>Select the protocol that the SLB instance uses when it performs health checks. For TCP listeners, both TCP health checks and HTTP health checks are supported.</p> <ul style="list-style-type: none">◦ A TCP health check implements detection at the network layer by sending SYN packets to check whether a port is open.◦ An HTTP health check verifies the health of a backend server by sending HEAD or GET requests to simulate browser access.
Health Check Method (for the HTTP and HTTPS health checks only)	<p>Health checks of Layer 7 (HTTP or HTTPS) listeners support both the HEAD and GET methods. The HEAD method is used by default.</p> <p>If your backend application server does not support the HEAD method or if the HEAD method is disabled, the health check may fail. To solve this issue, you can use the GET method instead.</p> <p>If the GET method is used and the response size exceeds 8 KB, the response is truncated. However, the health check result is not affected.</p> <div> Note Health checks of Layer 7 listeners of all regions support the GET method.</div>

Parameter	Description
Health Check Path and Health Check Domain Name (Optional) (for the HTTP health checks only)	<p>By default, SLB sends HTTP HEAD requests to the default homepage configured on the application server through the internal IP address of the backend ECS instance to perform health checks.</p> <p>If you do not use the default homepage of the application server for health checks, you must specify the path for health checks.</p> <p>Some application servers verify the host field in requests. In this case, the request header must contain the host field. If a domain name is configured in health check settings, SLB adds this domain name to the host field when SLB forwards a health check request to one of the preceding application servers. If no domain name is configured, SLB does not include the host field in requests and the requests are rejected by the application server, which may cause health checks to fail. If your application server verifies the host field in requests, you must configure a domain name in health check settings to ensure that the health check feature functions properly.</p>
Normal Status Code (for the HTTP health checks only)	<p>Select the HTTP status code that indicates successful health checks.</p> <p>Default values: http_2xx and http_3xx.</p>
Health Check Port	<p>The detection port used by the health check feature to access backend servers.</p> <p>By default, the backend port configured for the listener is used.</p> <div>  Note If a VServer group or a primary/secondary server group is configured for the listener, and the ECS instances in the group use different ports, leave this parameter empty. SLB uses the backend port of each ECS instance to perform health checks. </div>
Response Timeout	<p>The length of time to wait for a health check response. If the backend ECS instance does not send an expected response within the specified period of time, the health check fails.</p> <p>Valid values: 1 to 300. Unit: seconds. Default value for UDP listeners: 10. Default value for HTTP, HTTPS, and TCP listeners: 5.</p>
Health Check Interval	<p>The interval between two consecutive health checks.</p> <p>All nodes in the LVS cluster perform health checks independently and in parallel on backend ECS instances at the specified interval. The health check statistics of a single ECS instance cannot reflect the health check interval because the nodes perform health checks at different times.</p> <p>Valid values: 1 to 50. Unit: seconds. Default value for UDP listeners: 5. Default value for HTTP, HTTPS, and TCP listeners: 2.</p>
Unhealthy Threshold	<p>The number of consecutive failed health checks that must occur on a backend ECS instance before this ECS instance is declared unhealthy.</p> <p>Valid values: 2 to 10. Default value: 3.</p>

Parameter	Description
Healthy Threshold	<p>The number of consecutive successful health checks that must occur on a backend ECS instance before this ECS instance is declared healthy.</p> <p>Valid values: 2 to 10. Default value: 3.</p>
Health Check Requests and Health Check Results	<p>When you configure health check for a UDP listener, you can enter the request content (such as youraccountID) in the Health Check Requests field and the expected response (such as slb123) in the Health Check Results field.</p> <p>This operation adds the health check response logic to backend servers. For example, slb123 is returned when a youraccountID request is received.</p> <p>If SLB receives the expected response from a backend server, the health check succeeds. Otherwise, the health check fails. You can use this method to improve health check accuracy.</p>

8. Click **Next**.

Related information

- [Execute a health check script](#)

3. Execute a health check script

This topic describes how to use Cloud Assistant to execute a health check script on Elastic Compute Service (ECS) instances and view the health check results. Health check scripts are generated based on the health check configurations of listeners. You can manually execute a health check script after you attach backend servers to a Classic Load Balancer (CLB) instance.

Prerequisites

To execute a health check script on backend servers, make sure that the following requirements are met:

- Your CLB service is granted the required permissions to perform health checks on ECS instances. To grant the permissions, go to [RAM Roles](#).
- The backend servers that you want to check must be ECS instances that run Linux, the default Linux shell must be Bash, and Cloud Assistant must have been installed on the ECS instances. In addition, the ECS instances must be deployed in a virtual private cloud (VPC) and they must be in the Running state.
- Health checks are enabled for the listener of the CLB instance, and the ECS instances are added to the backend server group.

Context

Before you execute a health check script, take note of the following items:

- You cannot execute a health check script on backend servers that are associated with forwarding rules.
- The health check results returned after you execute a script may differ from those after the system automatically performs a health check. This is because different connections are used to perform these health checks. The health check results returned upon a script execution only provide you with suggestions on health check configurations. For backend server troubleshooting, the health check results returned upon scheduled health checks shall prevail.

Procedure

- 1.
- 2.
3. Find the CLB instance for which you want to check backend servers and click its ID.
4. On the **Listener** tab, find the listener that you want to manage and click **Modify Listener** in the Actions column.
5. On the **Configure Listener** page, click **Next** until the **Health Check** step appears.
6. Click **Health Precheck** in the **Advanced** section.
7. On the **Health Precheck** page, find the backend server on which you want to execute a health check script and click **Start Precheck** in the Actions column.

You can select up to five ECS instances at a time. If you want to execute the health check script on more than five ECS instances, divide these ECS instances into batches.

8. Click **OK** to execute the health check script. After the health check script is executed, the result is displayed in the console.

The following table describes the check items that are supported by listeners.

Listener type	Status of health check ports	iptables configuration	rpfilter configuration	Response upon HTTP probing	UDP probing
TCP	✓	✓	✓	✓	-
UDP	✓	✓	✓	-	✓
HTTP	✓	✓	✓	✓	-
HTTPS	✓	✓	✓	✓	-

To view the script execution result in details, log on to the [Cloud Assistant](#) console, select the region where the ECS instance is deployed, and then click the **Tasks** tab.

4. Disable the health check feature

This topic describes how to disable the health check feature for a Server Load Balancer (SLB) instance. If you disable the health check feature, requests may be distributed to unhealthy backend Elastic Compute Service (ECS) instances. This causes service disruptions. We recommend that you enable the health check feature.

Procedure

1. Log on to the **SLB console**.
2. On the **Instances** page, find the SLB instance that you want to manage and click its instance ID.
3. On the **Listener** tab, find the listener for which you want to disable the health check feature and click **Modify Listener** in the **Actions** column.
4. On the **Configure Listener** page, click **Next** to proceed to the **Health Check** wizard page.
5. Turn off the Enable Health Check switch and click **Next**.
6. Click **Submit** and click **OK**.

5. Health check FAQ

The following questions about health checks are frequently asked:

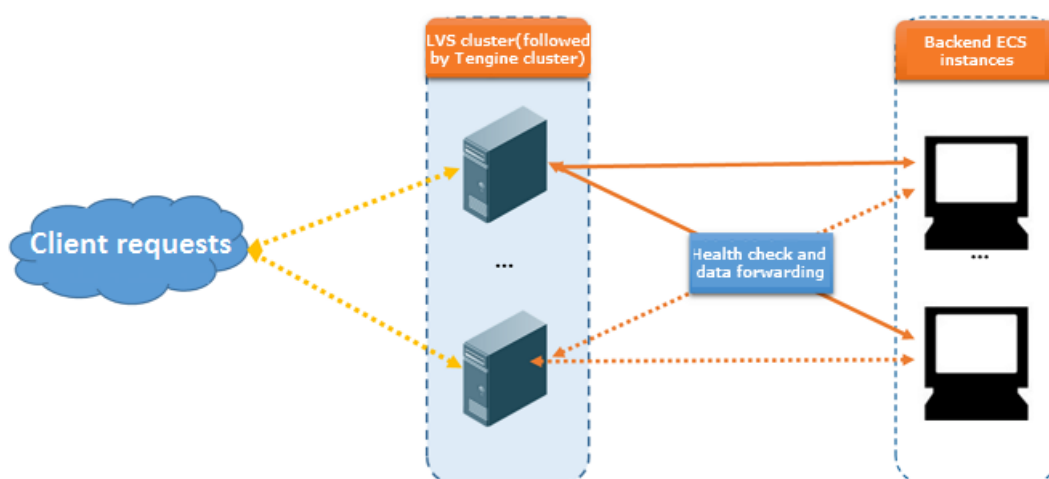
- [How does the health check feature of Server Load Balancer \(SLB\) work?](#)
- [What are the recommended configurations for health checks in SLB?](#)
- [Can I disable the health check feature?](#)
- [What is the recommended health check method for TCP listeners?](#)
- [How are health checks impacted if the weight of an ECS instance is zero?](#)
- [What health check method is used for HTTP listeners on backend ECS instances?](#)
- [What is the CIDR block that HTTP listeners use to Health Check to backend ECS instances?](#)
- [Why do the console and web logs display a different health check frequency?](#)
- [How do I handle a health check failure caused by a faulty backend database?](#)
- [Why is a network connection exception recorded in the backend service logs, but the TCP health check is displayed as successful?](#)
- [Why is the health check result returned as unhealthy even though the service is running normally?](#)

How does the health check feature of Server Load Balancer (SLB) work?

SLB performs health checks to verify the availability of backend ECS instances. If the health check feature is enabled and the check result shows that a backend ECS instance is unhealthy, the SLB instance does not forward new requests to the ECS instance till the instance becomes healthy.

SLB uses the CIDR block of 100.64.0.0/10 for health checks. Make sure that this CIDR block is permitted to access the backend ECS instances. You do not need to configure a security group rule to allow access from this CIDR block. However, if you have configured security rules such as iptables, you must allow access from this CIDR block. 100.64.0.0/10 is reserved by Alibaba Cloud. Other users cannot use any IP addresses within this CIDR block. Therefore, no relevant security risks exist.

For more information, see [Health check overview](#).



What are the recommended configurations for health checks in SLB?


To avoid impacts on system availability caused by frequent switching after failed health checks, the health check status switches only when health checks successively succeed or fail for a specified number of times within a certain time window. For more information, see [Configure health checks](#).

The following table describes the recommended health check configurations for TCP, HTTP, and HTTPS listeners.

Parameter	Recommended value
Response Timeout	5 seconds
Health Check Interval	2 seconds
Unhealthy Threshold	3


The following table describes the recommended health check configurations for UDP listeners.

Parameter	Recommended value
Response Timeout	10 seconds
Health Check Interval	5 seconds
Unhealthy Threshold	3
Healthy Threshold	3

 **Note** The recommended configurations help restore the service in a timely manner if the health check of a backend server fails. If you have higher requirements, you can specify a lower response timeout value. However, you must make sure the response time in the normal status is less than the timeout value that you have specified.

Can I disable the health check feature?

You can disable health checks only for HTTP and HTTPS listeners. Health check for TCP and UDP listeners cannot be disabled. For more information, see [Disable the health check feature](#).

 **Note** If you disable health checks, requests may be distributed to unhealthy ECS instances and cause impacts on your business. We recommend that you enable health checks.

What is the recommended health check method for TCP listeners?

For TCP listeners, both the HTTP and TCP health check methods are supported:

- A TCP health check sends SYN handshake packets to an instance to check whether the status of the instance is healthy.
- An HTTP health check simulates a process that uses a web browser to access resources by sending HEAD or GET requests to an instance and check whether the instance is healthy.

A TCP health check consumes less server resources. If the traffic load on backend servers is high, select TCP health checks. Otherwise, select HTTP health checks.

How are health checks impacted if the weight of an ECS instance is zero?

If you set the weight of an ECS instance to zero, SLB no longer forwards traffic to this ECS instance, and the ECS instance is determined as healthy in a health check.

After you set the weight of an ECS instance to zero, the ECS instance is removed from SLB. The weight is set to zero only when you restart or manage an ECS instance.

What health check method is used for HTTP listeners on backend ECS instances?

The HEAD method.

If you do not use the HEAD method for backend ECS instances, the backend ECS instances fail the health checks. We recommend that you access your own IP address on an ECS instance by using the HEAD method for testing. Run the following command on an ECS instance to access your IP address:

```
curl -v -O -I -H "Host:" -X HEAD http://IP:port
```

What is the CIDR block that HTTP listeners use to Health Check to backend ECS instances?

The CIDR block used by SLB health checks is . If backend ECS instances enable access control such as iptables, you must allow the access of on the internal network interface controller (NIC).

Why do the console and web logs display a different health check frequency?

Health checks are performed in clusters to avoid single points of failure. Proxies of SLB are deployed on multiple nodes. Therefore, the health check frequency recorded in web logs is different from the frequency configured in the console.

How do I handle a health check failure caused by a faulty backend database?

Problem description:

Two web sites are configured on an ECS instance. The website `www.test.com` is a static website, and the website `app.test.com` is a dynamic website. Both websites are configured with SLB services. A 502 error occurs due to a backend database fault when `www.test.com` is accessed.

Cause:

The domain name `app.test.com` is set for health checks. ApsaraDB RDS or self-built database failure causes an access error to `app.test.com` and the health check fails.

Solution:

Set the domain name that is used for health checks to `www.test.com`.

Why is a network connection exception recorded in the backend service logs, but the TCP health check is displayed as successful?

Problem description:

After a backend TCP port is configured in an SLB listener, the backend service logs frequently displays a network connection exception. The requests are sent from the SLB instance and the SLB instance also sends RST packets to the backend server at the same time.

```
java.io.IOException: Connection reset by peer
    at sun.nio.ch.FileDispatcherImpl.read0(Native Method)
    at sun.nio.ch.SocketDispatcher.read(SocketDispatcher.java:39)
    at sun.nio.ch.IOUtil.readIntoNativeBuffer(IOUtil.java:223)
    at sun.nio.ch.IOUtil.read(IOUtil.java:192)
    at sun.nio.ch.SocketChannelImpl.read(SocketChannelImpl.java:379)
    at io.netty.buffer.UnpooledUnsafeDirectByteBuf.setBytes(UnpooledUnsafeDirectByteBuf.java:446)
    at io.netty.buffer.AbstractByteBuf.writeBytes(AbstractByteBuf.java:871)
    at io.netty.channel.socket.nio.NioSocketChannel.doReadBytes(NioSocketChannel.java:225)
    at io.netty.channel.nio.AbstractNioByteChannel$NioByteUnsafe.read(AbstractNioByteChannel.java:115)
    at io.netty.channel.nio.NioEventLoop.processSelectedKey(NioEventLoop.java:507)
    at io.netty.channel.nio.NioEventLoop.processSelectedKeysOptimized(NioEventLoop.java:464)
    at io.netty.channel.nio.NioEventLoop.processSelectedKeys(NioEventLoop.java:378)
    at io.netty.channel.nio.NioEventLoop.run(NioEventLoop.java:350)
    at io.netty.util.concurrent.SingleThreadEventExecutor$2.run(SingleThreadEventExecutor.java:116)
    at java.lang.Thread.run(Thread.java:745)
```

Cause:

The problem is related to the health check mechanism.

TCP does not interrupt the upper-level services and is used to reduce the cost of health checks and the impacts on backend services. TCP health checks perform only a simple three-way handshake and then directly send RST packets to terminate the TCP connection. The following section describes the data exchange process:

1. The SLB server sends a SYN request packet to the backend SLB port.
2. The backend servers reply with a SYN-ACK package if the backend port is normal.
3. After the SLB instance receives the response from the backend port, the SLB instance determines that the listener and the backend servers are healthy.
4. The SLB instance sends a RST packet to the backend port to terminate the connection. A health check is complete.

After the health check succeeds, the SLB instance directly sends RST packets to terminate the connection. No data is sent afterwards. As a result, upper-level services such as the Java connection pool determine that the connection is abnormal and errors such as `Connection reset by peer` occur.

Solution:

- Change the protocol from TCP to HTTP.
- Filter the logs for requests from the CIDR block of the SLB server and ignore related error messages.

Why is the health check result returned as unhealthy even though the service is running normally?

Problem description:

The HTTP health check always fails, but the status code is normal. The `curl-I` command obtains the following status code:

```
echo -e 'HEAD /test.html HTTP/1.0\r\n\r\n' | nc -t 192.168.0.1 80
```

Cause:

If the returned status code is different from the healthy status code configured in the console, the backend ECS instances are declared as unhealthy. For example, assume that the configured healthy status code is `http_2xx`, the health check fails if the returned status code is not `http_2xx`.

No error occurred when a curl test is performed on the Tengine or Nginx cluster, but a 404 error occurred in the *test.html* test file because the default site is used in the echo test. The following figure shows the error details:

```
[root@iz28s-1-1-1-1 home]# echo -e "HEAD /test.html HTTP/1.0\r\n\r\n" | nc -t 10.161.93.136 80
HTTP/1.1 404 Not Found
Server: Tengine/2.1.0
Date: Mon, 16 Feb 2015 07:29:32 GMT
Content-Type: text/html
Content-Length: 585
Connection: close

[root@iz28s-1-1-1-1 home]# curl -I http://10.161.93.136/test.html
HTTP/1.1 200 OK
Server: Tengine/2.1.0
Date: Mon, 16 Feb 2015 07:29:41 GMT
Content-Type: text/html
Content-Length: 5
Last-Modified: Mon, 16 Feb 2015 07:27:00 GMT
Connection: keep-alive
ETag: "54e19bc4-5"
Accept-Ranges: bytes
```

Solution:

- Modify the main configuration file and comment out the default site.
- Add the domain name that is used for health checks in the health check configurations.