Alibaba Cloud Server Load Balancer

Health check

Issue: 20190909

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 due to product version upgrades , adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall 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 the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults " and "as available" basis. 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 bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, 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 content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, 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 the Alibaba Cloud website, product programs, or content 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 contact Alibaba Cloud directly if you discover any errors in this document.

Generic conventions

Table -1: Style conventions

Style	Description	Example
	This warning information 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.
	This warning information 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 restore business.
	This indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.	 Note: You can use Ctrl + A to select all files.
>	Multi-level menu cascade.	Settings > Network > Set network type
Bold	It is used for buttons, menus , page names, and other UI elements.	Click OK.
`Courier font`	It is used for commands.	Run the `cd /d C:/windows` command to enter the Windows system folder.
Italics	It is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	It indicates that it is a optional value, and only one item can be selected.	`ipconfig` *[-all|-t]*

Style	Description	Example
{} or {a|b}	It indicates that it is a required value, and only one item can be selected.	`swich {stand | slave}`

Contents

1 Health check overview

Server Load Balancer checks the service availability of the backend servers (ECS instances) by performing health checks. Health check improves the overall availability of the front-end service, and avoids impact on the entire service caused by exceptions of the backend ECS instances.

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

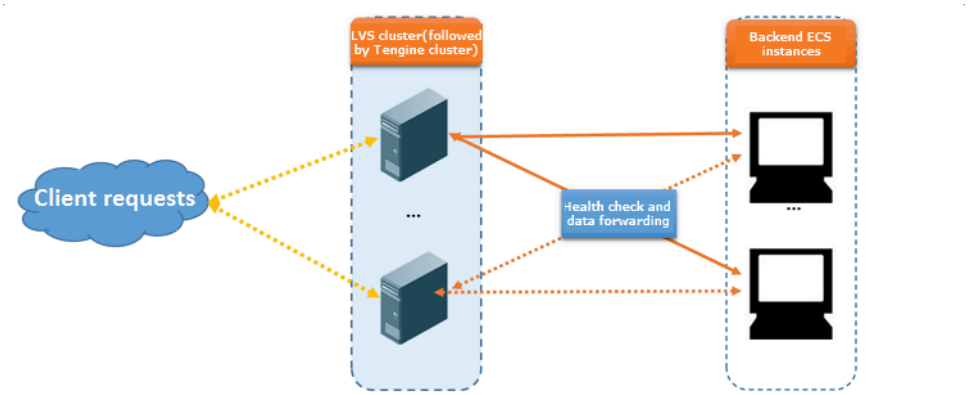
If your business is highly sensitive to traffic load, frequent health checks may impact normal service. You can reduce this impact by reducing the frequency of health checks, increasing the health check interval, or changing the HTTP health check to TCP health check. To guarantee the service availability, we do not recommend disabling all health checks.

Health check process

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

The node servers in the cluster independently perform health checks in parallel , according to the health check configuration. If a node server discovers an ECS instance is unhealthy, the node server will stop distributing requests to the ECS instance. This operation is synchronized through all node servers.

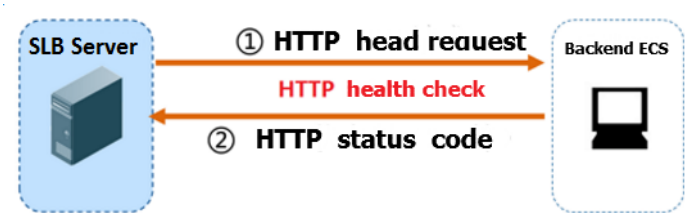
The IP address range used to perform the health check is 100.64.0.0/10. The backend servers cannot block this CIDR block. You do not need to additionally configure a security group rule to allow access from this CIDR block. However, if you have configured security rules such as iptables, allow access from this CIDR block (100.64. 0.0/10 is reserved by Alibaba Cloud, and other users cannot use any IP address in this CIDR block, so there is no security risk).



Health check of HTTP/HTTPS listeners

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

For HTTPS listeners, certificates are managed in SLB. Data exchange (including health check data and service interaction data) between SLB and backend ECS instances is not transmitted over HTTPS to improve the system performance.

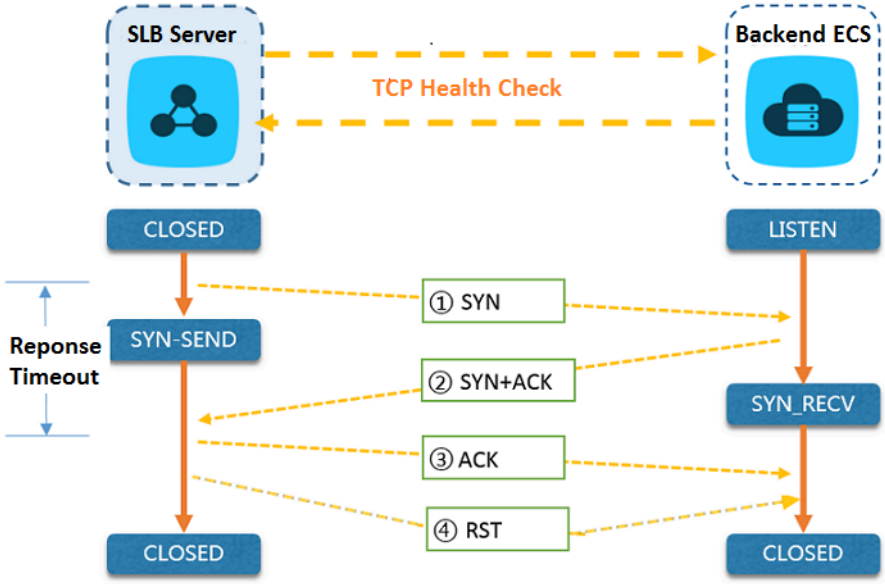


The health check process of a Layer-7 listener is as follows:

1. The Tengine node server sends an HTTP HEAD request to the intranet IP +Health Check Port+Health Check Path of the ECS instance 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 Tengine node server does not receive the response from the backend server within the Response Timeout period, the ECS instance is declared unhealthy.
4. If the Tengine node server receives the response from the backend ECS instance within the Response Timeout period, it compares the returned status code with the status code specified in the listener configuration. If the status code is the same, the backend server is declared healthy. Otherwise, the backend server is declared unhealthy.

Health check of TCP listeners

For TCP listeners, SLB detects the status of backend servers by sending TCP detections, 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 intranet IP + Health Check Port of the 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 required data packet from the backend server within the Response Timeout period, the ECS instance is declared unhealthy . Then, the LVS node server sends an RST data packet to the backend server to terminate the TCP connection.
4. If the LVS node server receives the data packet from the backend ECS instance within the Response Timeout period, the ECS instance is declared healthy. Then, the LVS node server sends an RST data packet to the backend server 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 an SYN + 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.

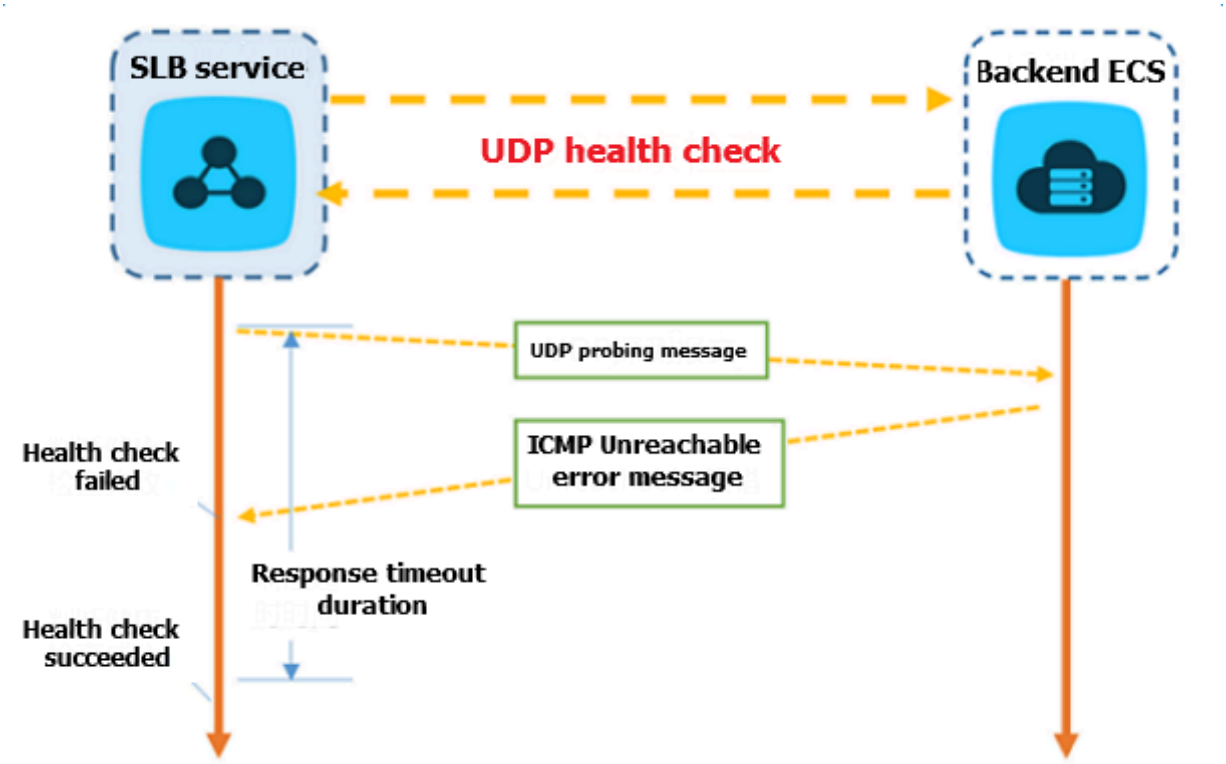
This process may make backend server think an error (such as an abnormal exit) occurred in the TCP connection and then throw a corresponding error message, such as `Connection reset by peer`.

Resolution:

- Use the HTTP health check.
- If obtaining real IP is enabled, ignore the connection errors caused by access of the SLB IP address.

Health check of UDP listeners

For UDP listeners, Server Load Balancer detects the status of the backend servers through UDP packet detection, 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 intranet IP + 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 will return 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 unhealthy.
4. If the LVS node server does not receive any messages within the Response Timeout period, the ECS instance is declared healthy.

> Note:
> For UDP health checks, the real status of the backend server and the health check result may not be the same in the following situation:
>
> If the ECS instance uses a Linux operating system, the speed of sending ICMP messages in high-concurrency scenarios is limited due to the anti-ICMP attack protection in Linux. In this case, even if an exception occurs in the ECS instance, SLB may declare the backend server healthy because the error message `port XX unreachabl e` is not returned. As a result, the actual service status is different from the health check result.
>
> Resolution:
>
> Set a pair of custom request and response for the UDP health check. If the custom response is returned, the ECS instance is considered healthy. Otherwise, the ECS instance is considered unhealthy. To achieve this, you must add corresponding configurations for the client.

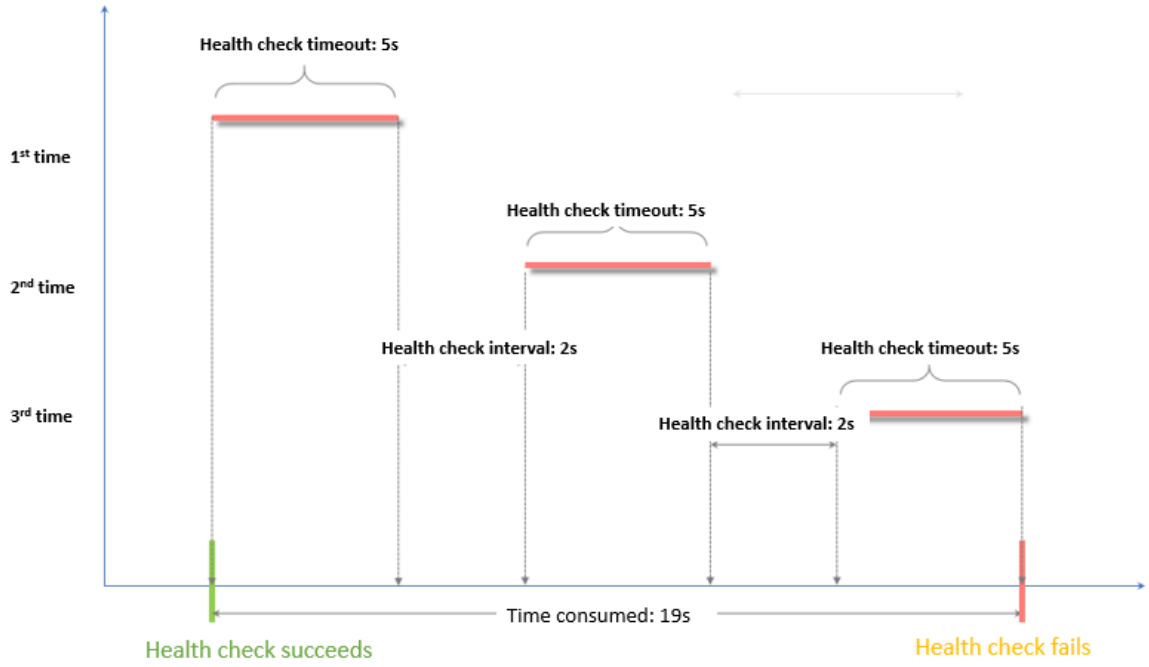
Health check time window

The health check function has effectively improved the availability of your business services. However, to reduce impact on the system availability caused by frequent system switches because of health check failure, SLB declares an ECS instance healthy or unhealthy only after successive successes or failures within a specified timeframe. 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 amount of time to wait for the response.)
- Health check threshold (The number of consecutive successful or failed health checks.)

The health check time window is calculated as follows:

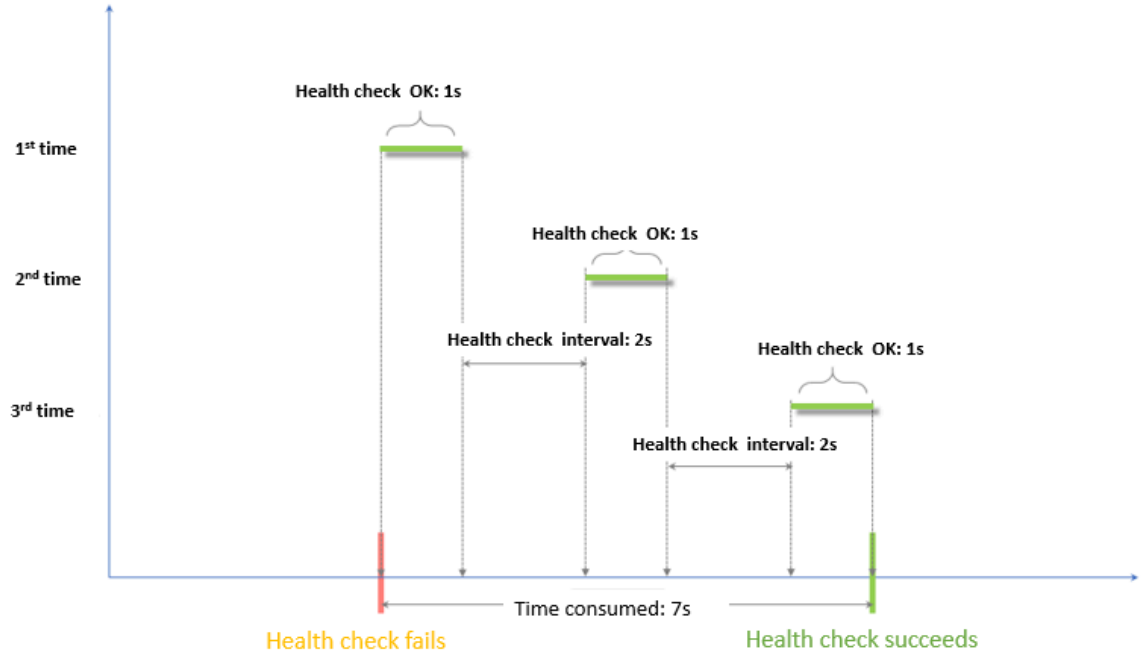
- Health check failure time window = Response Timeout x Unhealthy Threshold + Health Check Interval X (Unhealthy Threshold) -1)



- Health check success time window = (Response Time of a Successful Health Check X Healthy Threshold) + Health Check Interval X (Healthy Threshold-1)

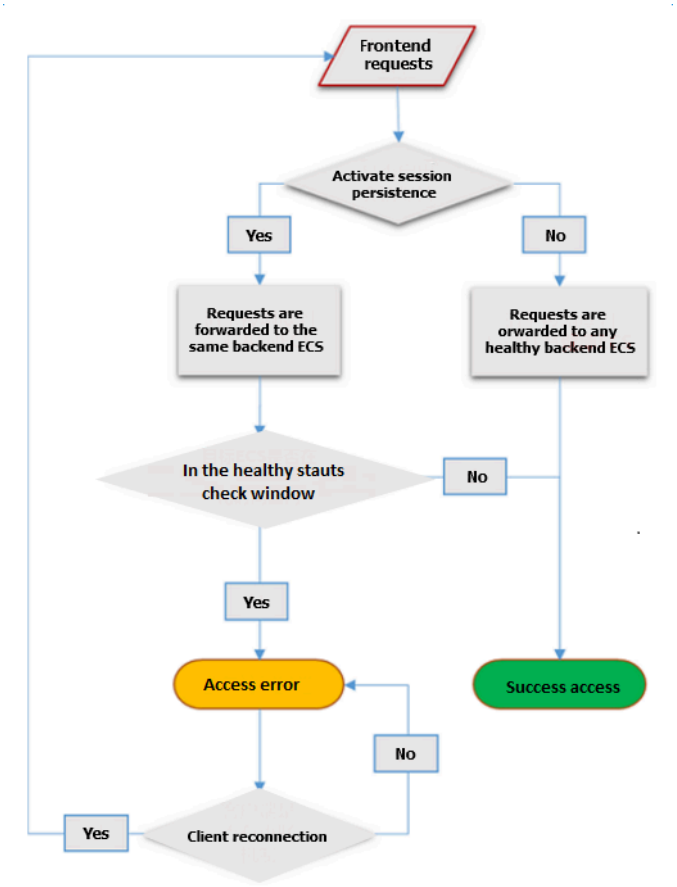
> Note:
> The success response time of a health check is the duration from the time when the health check request is sent to the time when the response is received. For TCP health check, the time is very short and almost negligible because TCP health check only detects whether the port is alive. For HTTP health check, 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 the requests forwarding:

- If the health check of the target ECS instance fails, new requests will not be distributed to the ECS instance. Therefore, there is no impact on the client access.
- If the health check of the target ECS instance succeeds, new requests will be 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. As a result, the client access fails.


Frontend requests
Activate session persistence
Yes
No
Requests are forwarded to the same backend ECS
Requests are orwarded to any healthy backend ECS
In the healthy stauts check window
No
Yes
Access error
Success access
No
Yes
Client reconnection

2 Configure health checks

You can configure the health check function when you add a listener. Generally, the default settings can meet your requirements.

Configure health checks

You can configure the health check function of a listener through the Server Load Balancer (SLB) console or APIs. For more information, see #unique_5 and Health check FAQ.

To configure the health check function, follow these steps:

1. Log on to the SLB console.
2. Select the region of the target SLB instance.
3. Find the target SLB instance and click the instance ID.
4. On the Instance Details page, click the Listeners tab.
5. Click Add Listener, or find the target listener and click Configure in the Actions column.
6. On the Health Check page, configure the health check function.

   We recommend that you use the default values when you configure the health check function.

   Table 2-1: Health check configurations

Configuration	Description
Health Check Protocol	For TCP listeners, both TCP health checks and HTTP health checks are supported. · TCP health checks are based on network layer detection. · HTTP health checks are performed by sending HEAD requests.

Configuration	Description
Health Check Method (HTTP and HTTPS health checks only)	Health checks of Layer-7 listeners (HTTP and HTTPS listeners) support both the HEAD and the GET request methods. The HEAD request method is used by default. Therefore, if your backend servers do not support the HEAD request method or the HEAD request method is disabled, health checks may fail. To resolve this issue, you can choose to use the GET request method for health checks. However, only the India (Mumbai) region supports the GET request method. Support for other regions is in development . Note: When the GET method is used, if the response length exceeds 8 KB, it is truncated, but the health check result is not affected.
Health Check Path and Domain Name (HTTP health checks only)	By default, SLB sends an HTTP HEAD request to the default homepage configured on the application server through the intranet IP address of the backend ECS instance to do health checks. If you do not use the default homepage of the application server to do health checks, you must specify the URL for health checks. Some application servers verify the host field in a request. Therefore, the request header must contain the host field. If a domain name is configured in the health check function, SLB adds the domain name to the host field when forwarding a request to the backend server. If no domain name is configured, no host field will be contained in the request, the health check request will be denied by the server, and the health check may fail. Therefore, if your application server verifies the host field in the request, you must configure a domain name to make sure the health check works.

Configuration	Description
Normal Status Code (HTTP health checks only)	Select the HTTP status code that indicates normal health checks. The default values are http_2xx and http_3xx.
Health Check Port	The detection port used by health checks to access backend servers. By default, the backend port configured in the listener is used. Note: If a VServer group or an active/standby 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 do health checks.
Response Timeout	The length of time to wait for the response from a health check. If the backend ECS instance does not send a correct response within the specified time, the health check fails. Value range: 1 to 300. Unit: seconds. Default value for UDP listeners: 10. Default value for HTTP, HTTPS, and TCP listeners: 5.
Health Check Interval	The time interval between two consecutive health checks. All node servers in the LVS cluster independently and concurrently perform health checks on backend ECS instances according to the interval. The statistics from a health check request on a single ECS instance cannot reflect the health check interval because the health check time of each node server is not synchronized. Value range: 1 to 50. Unit: seconds. Default value for UDP listeners: 5. Default value for HTTP, HTTPS, and TCP listeners: 2.

Configuration	Description
Unhealthy Threshold	The number of consecutive failures of health checks performed by the same LVS node server on the same ECS instance before the ECS instance is declared as unhealthy (from success to failure). Value range: 2 to 10. Default value: 3.
Healthy Threshold	The number of consecutive successes of health checks performed by the same LVS node server on the same ECS instance before the ECS instance is declared as healthy (from failure to success). Value range: 2 to 10. Default value: 3.
Health Check Requests and Results	When you configure health checks for UDP listeners, you can enter the request contents (such as youraccountID) in Health Check Request and the expected response (such as slb123) in Health Check Response. Add the corresponding health check response logic to the application logic of the backend server. For example, return slb123 when youraccountID is received. If SLB receives the expected response from the backend server, the health check succeeds. Otherwise, the health check fails. This method can guarantee the reliability of health checks.

7. SLB supports health check diagnostics on backend servers.
   a. Click Health Check Diagnostics in the Advanced section.

      Note:
      If you log on as a RAM user, you must grant permissions to the RAM user before it can use the health check diagnostics function. Click here to grant permissions to a RAM user.

   b. Find the target backend server, and click Diagnose in the Actions column.

      To check multiple backend servers at a time, select target backend servers and click Batch. Up to five backend servers can be selected at the same time. If the

number of backend servers is larger than five, check the backend servers in batches.

c. Click OK.

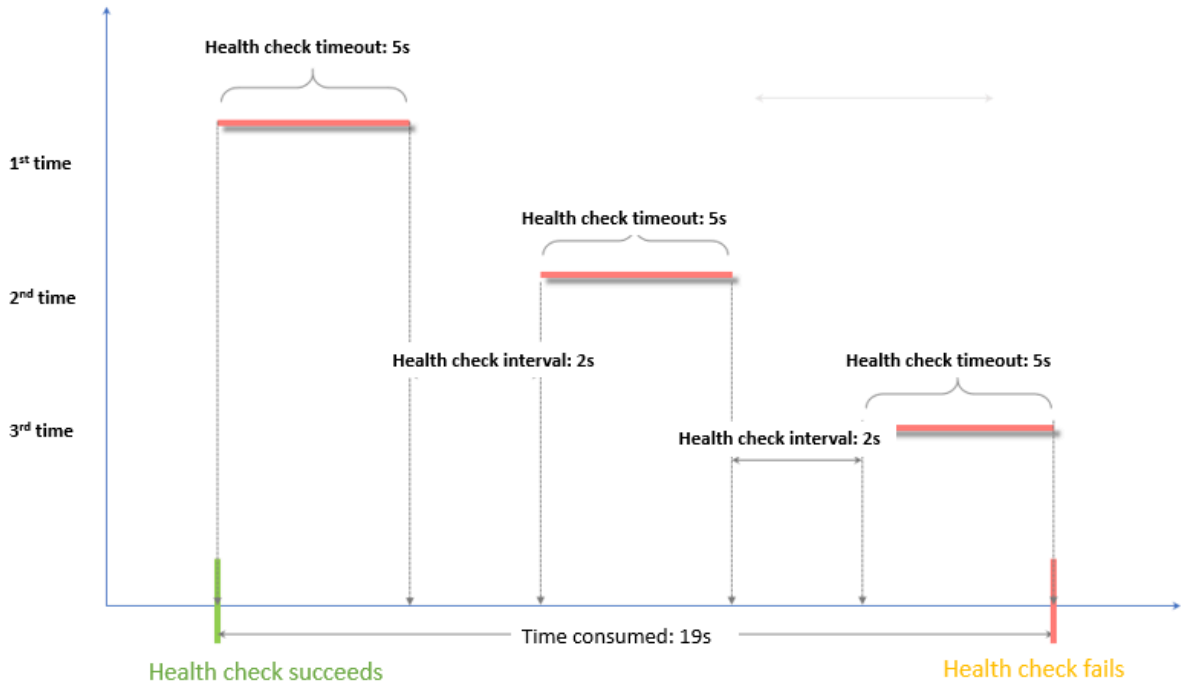
Example of the health check response timeout and health check interval

Take the following health check configurations as the example:

- Response Timeout: 5 seconds
- Health Check Interval: 2 seconds
- Healthy Threshold: 3 times
- Unhealthy Threshold: 3 times

Health check failure time window = Response Timeout × Unhealthy Threshold + Health Check Interval × (Unhealthy Threshold - 1). That is, 5 × 3 + 2 × (3 - 1) = 19s.

The following figure shows the process to declare an unhealthy backend server:



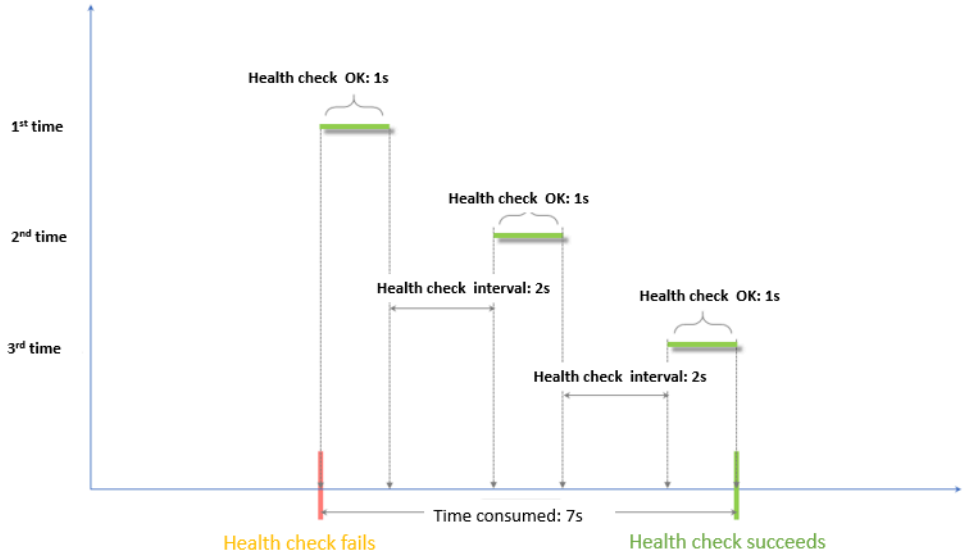
Health check success time window = Health check response time × Healthy Threshold + Health Check Interval × (Healthy Threshold - 1). That is, (1 × 3) + 2 × (3 - 1) = 7s.

Note:

Health check response time is the duration from the time when the health check request is sent to the time when the response is received. When the TCP health check

is used, the time is very short and almost negligible because the health check only detects whether the port is alive. When the HTTP health check is used, the response time depends on the performance and load of the application server and is usually within seconds.

The following figure shows the process to declare a healthy backend server (assume that it takes one second for the backend server to respond to the health check request):



Configure a domain name in HTTP health checks

When the HTTP health check is used, you can set a domain name for the health check, but it is not required. Some application servers verify the host field in the request. Therefore, the request header must contain the host field. If a domain name is configured in the health check function, SLB adds the domain name to the host field when forwarding the request to the backend server. If not, the health check request will be denied by the server and the health check may fail. Therefore, if your application server verifies the host field in the request, you must configure a domain name to make sure the health check works.

3 Disable the health check function

If you disable the health check function, requests may be distributed to unhealthy ECS instances, resulting in disruption to your services. Therefore, we recommend that you enable the health check function.

Context

Note:

You can only disable the health check function for HTTP and HTTPS listeners. The health check function for UDP and TCP listeners cannot be disabled.

Procedure

1. Log on to the SLB console.
2. Select the region of the target SLB instance. On the Server Load Balancer page, find the target SLB instance and click the instance ID.
3. On the Listeners tab, find the target listener and click Configure in the Actions column.
4. On the Configure Listener page, click Next until the Health Check tab is displayed.
5. Turn off Enable Health Check, click Next, click Submit, and then click OK.

4 FAQ

4.1 How do I troubleshoot health check exceptions of a Layer-4 (TCP/UDP) listener?

The health check function is used to determine whether your backend servers are normal. When a health check exception occurs, it generally means that your backend server is abnormal. The exception may also be caused by incorrect health check configurations. This topic describes how to troubleshoot a health check exception of a Layer-4 (TCP/UDP) listener.

Procedure

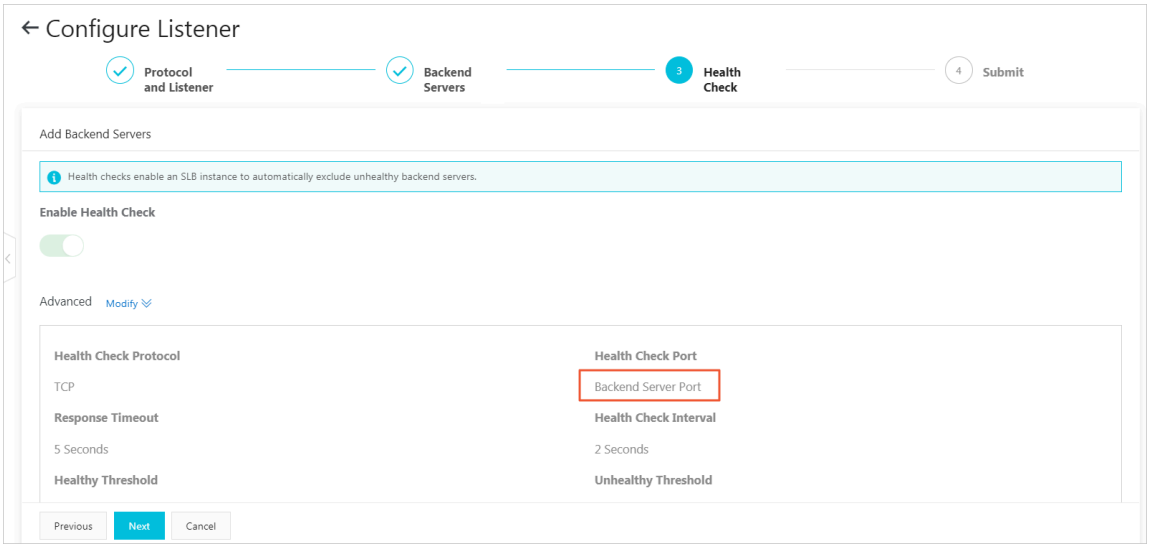
1. Ensure that the backend server does not block the CIDR block 100.64.0.0/10 through iptables or other third-party firewalls or security software.

   The SLB instance communicates with backend servers by using IP addresses in the reserved CIDR block 100.64.0.0/10. If the CIDR block is blocked, a health check exception occurs and the SLB instance cannot work normally.

2. Run the `telnet` command to test the backend server.

   a) Log on to the SLB console and check the health check configurations.

   By default, the port of the backend server is used as the Health Check Port. You can also set the port manually. In this example, the port of the backend server, namely port 80, is used.

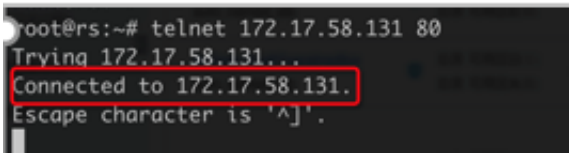


   b) Run the following command to connect to the health check port. The health check port configured on the SLB instance must be the same as the listening port on the backend server.

```
telnet  172 . 17 . 58 . 131  80
```

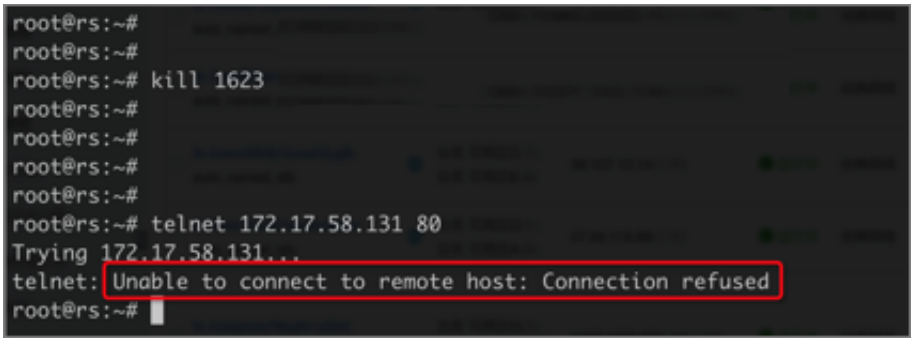
   In this example, *172.17.58.131* is the intranet IP address of the backend server, and *80* is the health check port. By default, the port of the backend server is used as the health check port. You can configure the health check port according to your actual situation.

   - In normal conditions, `Connected to xxx . xxx . xxx . xxx` is returned. This indicates that the port on the backend server is working (listening) normally and the health check is normal, as shown in the following figure.



   - Exception example: Assume you do not change the listener configurations of the SLB instance but stop the listening process of port 80 on the backend

server. Then, if you run the `telnet` command, the system prompts that the host cannot be connected. This means that a health check exception occurs if the listening process of port 80 stops, as shown in the following figure.



3. Optional: Layer-4 listeners also support HTTP health checks. To use HTTP health checks, see #unique_10. The method for troubleshooting HTTP health check exceptions is the same for Layer-4 listeners and Layer-7 listeners.

4.2 Troubleshoot a health check exception of a Layer-7 listener (HTTP/HTTPS)

The health check function is used to determine whether your backend servers are normal. If a health check exception occurs, it generally means that your backend server is abnormal. The exception may also be caused by incorrect health check configurations. This topic describes how to troubleshoot a health check exception of a Layer-7 (HTTP/HTTPS) listener.

Procedure

1. Ensure that the backend server does not block the CIDR block 100.64.0.0/10 through iptables or other third-party firewalls or security software.

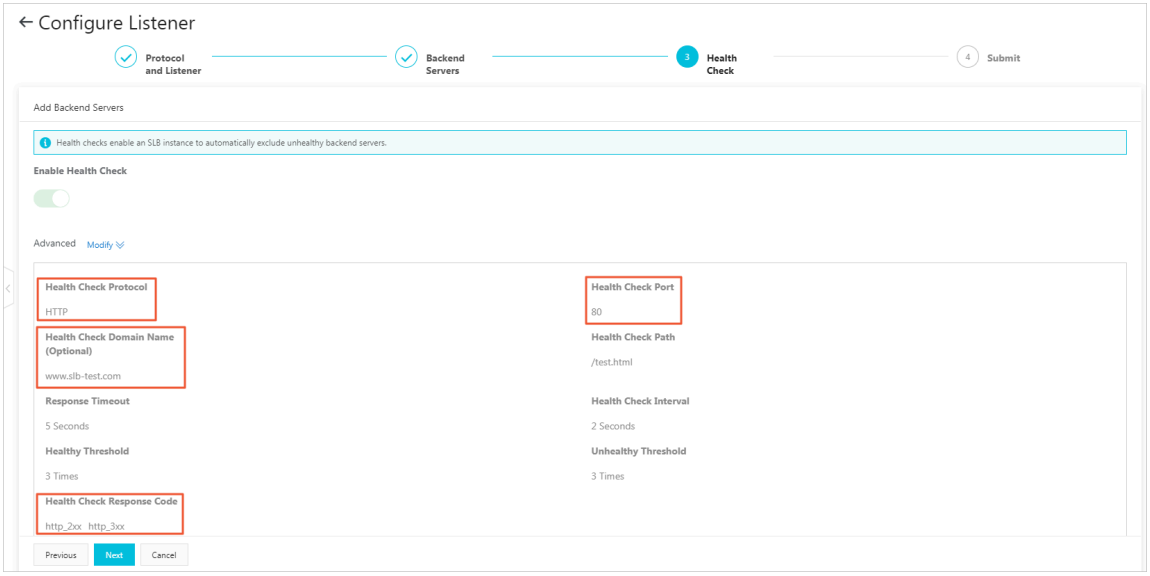
   The SLB instance communicates with backend servers by using IP addresses in the reserved CIDR block 100.64.0.0/10. If the CIDR block is blocked, a health check exception occurs and the SLB instance cannot work normally.

2. Access the HTTP service on the backend server from the backend server to ensure that the HTTP service works normally.

   a) Log on to the SLB console and click the ID of the target SLB instance. On the Listeners tab page, click Configure in the Actions column of the target listener. Then you can view the health check configurations.

   In this example, an HTTP listener is used and the intranet IP of the backend server with the health check exception is 10.0.0.2. Other health check configurations are as follows:

   - Health Check Port: 80
   - Health Check Domain Name: `www . slb - test . com`
   - Health Check Path: `/ test . html`



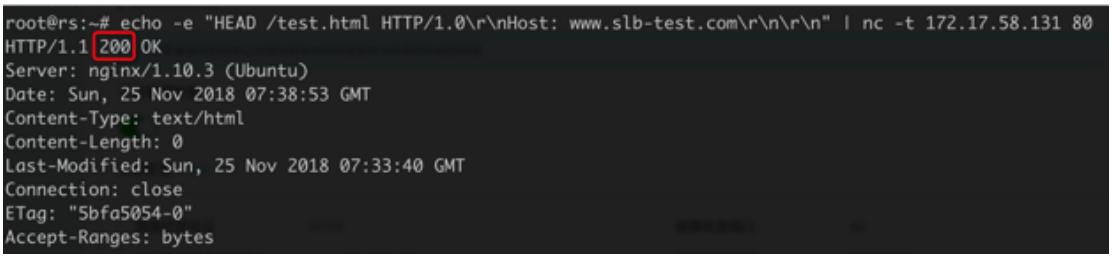
   b) The following example uses a Linux environment. Run the `nc` or `curl` command to test the HTTP service on the backend server. Ensure that the configurations of health check path, health check port, and health check

domain name are the same for the HTTP service and the backend server. Otherwise, a health check exception occurs.

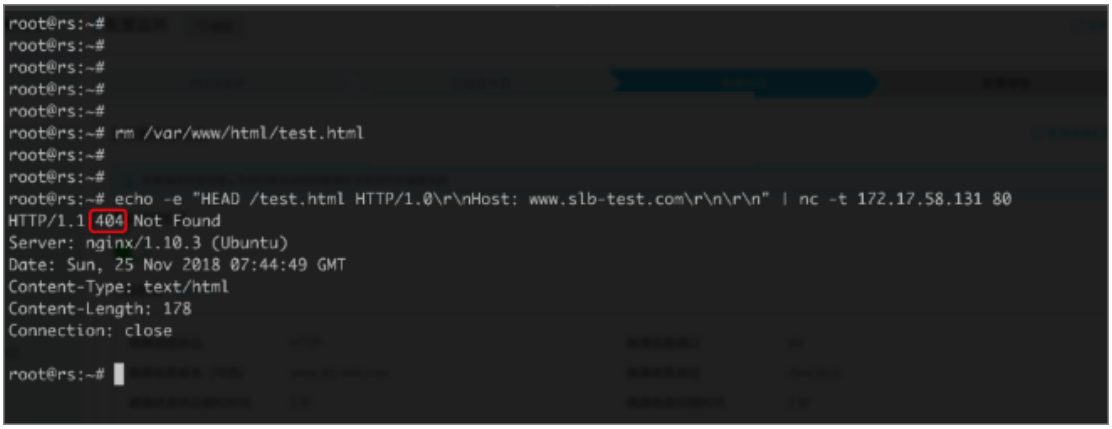
In this example, the `nc` command is used. Configure the health check path, health check domain name, internet IP address, and health check port according to your actual situation.

```
echo -e "HEAD /test.html HTTP/1.0\r\nHost: www.slb-test.com\r\n\r
\n" | nc -t 172.17.58.131 80
```

- In normal conditions, `200` or `2xx / 3xx` is returned, as shown in the following figure.



- Exception example: Assume you do not change the listener configurations of the SLB instance but delete the /test.html page on the backend server. Then, when you run the `nc` command, the error code 404, instead of 2xx or 3xx, is returned, indicating a health check exception has occurred, as shown in the following figure.



4.3 Health check FAQ

The following are frequently asked questions about health checks:

- How does the health check function of Server Load Balancer (SLB) work?
- What are the recommended configurations for health checks in SLB?

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

SLB checks the service availability of backend servers (ECS instances) by performing health checks on backend servers. When SLB detects that an ECS instance is unhealthy, SLB stops distributing requests to the ECS instance until it becomes healthy again.

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 need to 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 there is no security risk.)

For more information, see Health check overview.

What are the recommended configurations for health checks in SLB?

To avoid the impact of backend server switching caused by frequent health check failures on system availability, health check failures or successes must reach a certain threshold before the health check status of a backend server is switched.

The following are recommended health check configurations for TCP, HTTP, and HTTPS listeners.

Configuration	Recommended value
Response timeout	5 seconds
Health check interval	2 seconds
Unhealthy threshold	3 times

The following are recommended health check configurations for UDP listeners.

Configuration	Recommended value
Response timeout	10 seconds
Health check interval	5 seconds
Unhealthy threshold	3 times
Healthy threshold	3 times

Note:
These configurations are conducive to restoring the service when 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 function?

You can only disable health checks for HTTP and HTTPS listeners. Health checks for UDP and TCP listeners cannot be disabled.

Note:
If the health check function is disabled, requests may be distributed to unhealthy ECS instances, which can lead to service interruptions. Therefore, we recommend that you enable health checks.

What is the recommended health check method for TCP listeners?

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

- TCP health checks send SYN handshake packets to backend servers to check whether the ports of backend servers are normal.

- HTTP health checks detect the health status of applications on backend servers by sending HEAD and GET requests to simulate visits from the browser of a user.

The TCP health check minimally impacts the performance of backend servers and consumes less server resources. Select TCP health check if the traffic load on backend servers is high, and select HTTP health check if not.

Is there any impact to health checks if the weight of an ECS instance is zero?

If you set the weight of an ECS instance to zero, SLB will no longer forward traffic to this ECS instance and health checks for Layer-4 listeners will indicate abnormal of backend ECS instances (the health check is normal for Layer-7 listeners).

Setting the weight value to zero is equal to manually removing the ECS instance from SLB. Generally, the weight is set to zero only when you restart, adjust, or maintain the ECS instance.

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

HEAD request method.

If you disable the HEAD request method for backend ECS instances, health checks on the backend ECS instances will fail. We recommend that you access your own IP address on the ECS instance by using the HEAD method for testing:

```
curl -v -0 -I -H " Host :" -X HEAD http :// IP : port
```

What are the ranges of IP addresses that HTTP listeners use to perform health checks on backend ECS instances?

The IP address range used by SLB health checks is 100.64.0.0/10（100.64.0.0/10 is reserved by Alibaba Cloud, and will not be used by any user, there is no security risk）. If the backend ECS instance enables access control such as iptables, you need to allow the access of 100.64.0.0/10（100.64.0.0/10 is reserved by Alibaba Cloud, and will not be used by any user, there is no security risk）on the intranet NIC.

Why is the health check frequency that is displayed on the console different from that recorded in the web logs?

Health checks are performed in the cluster to avoid single points of failure. Therefore, the health check frequency recorded in the logs is different from the frequency configured in the console.

Do health checks use system resources?

HTTP health checks consume few resources of the backend ECS instances.

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

Symptoms:

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. A 502 error occurs due to a backend database fault when accessing www.test.com.

Cause:

The domain name app.test.com is configured for health checks. RDS or self-built database failure causes the access error to app.test.com. Therefore, the health check fails.

Solution:

Configure the domain name 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?

Symptoms:

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

Cause:

The problem is related to the health check mechanism.

TCP is transparent to the upper-Layer applications and is utilized to reduce the cost of health checks and the impact on backend service. TCP health checks only perform a simple three-way handshake and then directly send RST packets to terminate the TCP connection. The data exchange process is as follows:

1. The SLB instance sends a SYN packet to the backend port.
2. The backend server replies with a SYN-ACK if the backend port is normal.

3. After successfully receiving the response from the backend port, the SLB instance considers that the port is in normal status and the status of the backend server is normal.
4. The SLB instance sends a RST packet to the backend port to actively terminate the connection. For now, a health check is completed.

After the health check succeeds, the SLB instance directly sends RST packets to terminate the connection and no data is sent afterwards. Therefore, upper-Layer services (such as Java connection pool) deem that the connection is abnormal and errors such as `Connection reset by pee` occur.

Solution:

- Use the HTTP protocol.
- In terms of the service, filter the logs from the SLB IP address range and ignore related error messages.

Why is the health check result returned as abnormal when the service is running normally?

Symptoms:

The HTTP health check always fails, but the status code obtained by performing the `curl -l` test is normal as follows:

```
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 normal status code configured in the console, the backend ECS instance is declared as unhealthy. For example, if the configured normal status code is http_2xx, all other status codes returned not matching this status code will be considered as health check failure.

No error occurred when a curl test is performed on the Tengine/Nginx cluster, but a 404 error occurred in the *test.html* test file because the default site is used in the echo test.

Solution:

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