

Tencent Cloud EdgeOne Origin Configuration Product Documentation





Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice

🔗 Tencent Cloud

All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

Origin Configuration

Load Balancing

Overview

Quickly Create Load Balancers

Health Check Policies

Viewing the Health Status of Origin Server

Related References

Load Balancing-Related Concepts

Introduction to Request Retry Strategy

Origin Group Configuration

Origin-pull configuration

Origin-Pull Timeout

Configuring Origin-Pull HTTPS

Host Header Rewrite

Controlling Origin-pull Requests

Redirect Following During Origin-Pull

HTTP/2 Origin-Pull

Range GETs

Related References

Id Version Origin Group Compatible Related Issues

Origin Configuration Load Balancing Overview

Last updated : 2024-05-29 10:33:37

EdgeOne Load Balancing is ideal for scenarios where high availability of origins is crucial. It supports the configuration of multi-level secondary sources for disaster recovery switching. It can proactively probe the health status of origins. This proactive measure blocks failed origins and directs business traffic to healthy origins.

Note:

EdgeOne Load Balancing feature is in beta testing. If you want to use it, you can Contact Us.

Use Cases

Hardware failures, network failures, configuration errors, security attacks, natural disasters, human errors, and other unforeseen circumstances, can affect the availability of the origin. For businesses that require high availability, such as finance, gaming, audio and video, and e-commerce, even brief failures of the origin can result in significant losses.

Therefore, it is necessary to implement primary/secondary disaster recovery and health checks for the origin. **Primary/Secondary Disaster Recovery:** When the primary source becomes unavailable, it is automatically

switched to the secondary source to ensure business continuity.

Proactively Checking origin Health Status: Preemptively disables failed origins, and redirects business traffic to healthy origins. Prevents a situation where a significant number of legitimate service requests are still directed to malfunctioning origins in the event of a failure.

Supported Capabilities

1. Supports the configuration of multi-level secondary sources for multi-source disaster recovery.

2. Supports the configuration of health check policies such as ICMP Ping, HTTP/HTTPS, TCP, and UDP to preemptively disable failed origins and redirect business traffic to healthy origins.

3. Provides a fallback retry policy. It retries directing the traffic to alternative healthy origins when real business traffic requests fail.





More Information

Quickly Create Load Balancers Load Balancing-Related Concepts Health Check Policies

Quickly Create Load Balancers

Last updated : 2024-05-29 10:33:37

This document guides you on how to create a Cloud Load Balancer instance.

Note:

EdgeOne Load Balancing feature is in beta testing. If you want to use it, you can Contact Us.

Sample Scenario

For example, you currently have an acceleration domain www.example.com , with three origins 1.2.3.4 , 2.3.4.5 , and 3.4.5.6 . Under normal circumstances, both 1.2.3.4 and 2.3.4.5 are used as primary origins. You have already configured them as the origin group named primary_origins following the Origin Group Operation Guide. The server 3.4.5.6 is used as a standby origin in a group called backup_origins, which is only used when the primary origins fail. In cases where a real business request fails, retries are attempted with other healthy servers within the same group. Additionally, there is a requirement for proactive probing to actively identify and disable unhealthy origins.

Directions

1. Log in to the Tencent Cloud EdgeOne console. In the left menu bar, click the **Site List**. Within this list, click the **Site that** need to be configured to go to the details page.

2. On the site details page, click **Origin Settings** > Load Balancing.

3. On the Load Balancing page, click **Create Instance**.

Create Instance				Please enter the instance name
Load balancer/ID	Instance type	Health Check Policy	Traffic scheduling policy	Status
			No data yet	
Total items: 0				10 💌 / page

4. Proceed to step 1 of choosing the origin. You need to fill in the instance name, choose the instance type, and add an origin group.

Taking this scenario as an example, add the origin group primary_origins as a priority 1 origin group, add the origin group backup_origins as a priority 2 origin group, and click **Next**.

nstance name	load_balancer					
	1-200 characters, allowe	ed characters are a-z, A-Z, 0-9, _, -				
nstance type	O HTTP-specific type	General Type				
Add origin group	Priority	Origin Group		Origin type	Origin group information	
	1	primary_origins	▼	HTTP-specific type	1.2.3.4(50.00%) 2.3.4.5(50.00%)	
	2	backup_origins	▼	HTTP-specific type	3.4.5.6	

Parameter	Description
Instance name	Limit to 1-200 characters in length. Allowed characters are a-z, A-Z, 0-9, _,
Instance type	HTTP-specific Type: Supports adding both HTTP-specific and general origin groups. It is only applicable for reference by site acceleration-related services, such as domain services and rule engines.General Type: Only supports adding general origin groups. It is applicable for site acceleration services including domain services and rule engines, and reference by L4 proxy service.
Add origin group	In the CLB instance, the smallest configuration dimension for an origin is the origin group. You need to configure the origin into an origin group and add it here. For more details, see Origin Group Configuration. You can set priorities for the added origin groups. Traffic will not be directed to origins in lower-priority origin groups if there are healthy origins in higher-priority origin groups. Up to 10 origin groups can be configured, with lower numbers indicating higher priorities.

5. Proceed to step 2 of health check policy. It supports four types of probes: ICMP Ping, HTTPS/HTTP, TCP, and UDP. Tencent Cloud EdgeOne will actively send probe requests to your origin to check its latency and health status. You can choose the appropriate probe frequency based on the load condition of your origin. Here, choose ICMP Ping as the probe policy. For a detailed introduction to probe policy configuration, see Introduction to Health Check Policy. After configuration is completed, click **Next**.

	ect the latency and	health of your origin by proactively sending probi	ng requests to your origin based on the following	g configuration you select.
etection Policy				
ICMP Ping		HTTPS/HTTP	ТСР	UDP
Only check network of host accessibility	connectivity,	Suitable for applications that need to recognize the content of the request, such as Web applications, app services, etc.	Suitable for scenarios that require high reliability and data accuracy and low transmission speed, such as file transfer, remote login, etc.	Suit for scenarios that require high transmission efficiency and relatively low accuracy, such as instant messaging, online video, etc.
asic configuration	1			
etection frequency	Every 30 secon	ds	▼	
xpand Advanced Confid	guration 🗸			

Note:

If you do not want EdgeOne's nodes to initiate any probe requests to origins, you can choose **Not Enabled**. In this case, the Load Balancing instance will default to traffic scheduling based on the priority order of the origin groups from step 1. If a request to a particular origin fails 5 times within 60 seconds, the corresponding origin will be disabled for 10 minutes according to the default policy.

Using this policy **will not be able to disable the origin of the failure in advance, and it will not be able to automatically and quickly recover the traffic scheduling after the origin returns to normal**. Compared with enabling active probe, using this policy may cause you to encounter more failed requests during the origin failure period. Therefore, if you want your business to have higher availability, it is recommended that you enable active probe.

6. Proceed to step 3 of traffic scheduling policy. The current traffic scheduling policy defaults to failover based on the priority order according to the results of active probes. When real business requests fail to retrieve content from the origin during the backsource process, support for request retry is available. There are two request retry policies available. For details, see Introduction to Request Retry Policy.

Policy 1: When a real business request fails to access a certain origin, it directly retries with another origin within the next lower priority origin group. This is suitable for scenarios where the performance of both origin group 1 and origin group 2 is similar.

Policy 2: When a real business request fails to access a certain origin, it directly retries with another origin within the same priority origin group. This is suitable for scenarios where the performance of origin group 1 is significantly better than that of origin group 2.



raffic scheduling policy	Failover in order of priority				
	EdgeOne will actively probe the cor	figured origins based on the health check policy	ou have set, block faulty origin grou	ips in the order of priorit	y, and route
Request retry policy	The process of origin pulling may fa	il due to network fluctuations or other reasons, th	erefore two retry strategies are prov	ided.	
	Strategy 1		O Strategy 2		
	When a real request fails to access current priority origin group. Applic origin group 2.	a specific origin , it will directly retry to other orig able scenario: Similar performance between origir	ns in the When a real reque group 1 and current priority ori than origin server	st fails to access a speci gin group. Applicable sco group 2.	fic origin , it enario: Origii
	Origin 1	Origin 2	Origi	n 1	Oriç
	·····❸→ [●] Origin Retry the next of the	e origin in origin group Origin	— ⊗ → 👻 (Prigin Retry another	>
	Irigin	Origin	e	Origin (ame
	Irigin	Origin	e	Drigin	2

7. Taking this sample scenario as an example, policy 2 can be chosen. Click **Complete** to finish creating the instance.

Health Check Policies

Last updated : 2024-05-29 10:33:37

This document introduces the probe methods and their principles within health checks, the origin health determination criteria and the calculation methods.

Note:

EdgeOne Load Balancing feature is in beta testing. If you want to use it, you can Contact Us.

Principle of Health Checks

After configuring health check policies, EdgeOne's probe nodes in different regions will send probe requests to your origin and determine the health status of the origin based on the response results. Health check policies consist of probe methods and origin health determination criteria. The probe method determines the type of probe request, while the origin health determination criteria determine how the response results are processed.



Probe Method

Currently, supports for ICMP Ping, HTTP/HTTPS, TCP, and UDP as the four methods of probe. For more details, see The Principle Introduction of Probe Methods. The following are the explanations for the corresponding configuration items:

Probe Method	Applicable Scenario	Configuration Item	Description
ICMP Ping	Only probes network connectivity, and host reachability.	Probe Frequency	Required, with optional intervals of every 30 seconds, every 60 seconds, every 3 minutes, every 5 minutes, or every 10 minutes.

HTTP/HTTPS	Applicable for applications that require content recognition in	Probe Frequency	Required, with optional intervals of every 30 seconds, every 60 seconds, every 3 minutes, every 5 minutes, or every 10 minutes.
	requests, such as web applications and app services.	URL	Required, the full URL for health checks, for example: <pre>www.example.com/test</pre> .
		Probe Port	Required, defaulting to port 80. It is recommended not to modify this unless a specific port needs to be designated.
		HTTP Method	Required, the HTTP method for health checks is by default HEAD, with options including GET or HEAD. If the HEAD method is used, the server returns only HTTP header information, which can reduce backend overhead and enhance request efficiency. The corresponding origin service must support HEAD. If the GET method is used, the origin service simply needs to support GET.
		HTTP Status Code	Required, the origin is considered healthy when the status code matches the selected status codes. By default, this includes 2XX, with options to select: 1XX, 2XX, 3XX, 4XX, 5XX.
		Follow Redirects	Disabled by default. When enabled, the probe node will initiate another probe based on the 301/302 redirect address responded by the origin. It Uses the status code of the final redirection response as the determination result for the health status. Up to 3 redirects are supported.
		Custom Request Headers	Optional, custom request headers can be configured to be sent with the health check requests to the origin, with a maximum of 8 configurations allowed, for example: host: www.example.com .
ТСР	Suitable for scenarios where high reliability and data accuracy are	Probe Frequency	Required, with optional intervals of every 30 seconds, every 60 seconds, every 3 minutes, every 5 minutes, or every 10 minutes.
	essential, but	Probe Port	Required, defaulting to port 80. It is

	transmission speed is of lesser importance, such as file transfers and remote log-ins.		recommended not to modify this unless a specific port needs to be designated.
UDP	Suitable for scenarios	Probe Frequency	Required, with optional intervals of every 30 seconds, every 60 seconds, every 3 minutes, every 5 minutes, or every 10 minutes.
	where high transmission efficiency is crucial and a relatively lower level of accuracy is acceptable, such as instant messaging and online video streaming.	Probe Port	Required, defaulting to port 80. It is recommended not to modify this unless a specific port needs to be designated.
		Probe Request	Required, customize the content of the health check request, with a limit of 500 characters.
		Probe Response Result	Required, customize the content of the health check request, with a limit of 500 characters.

Origin Health Determination Criteria

Choose any of the probe policies: ICMP Ping, HTTP/HTTPS, TCP, and UDP. Click **Show Advanced Configuration** to configure origin health determination criteria. The following are the descriptions for each configuration item:

	Select origin		2 Health Check Policy	
i EdgeOne will dete	ect the latency and	health of your origin by proactively sending probi	ng requests to your origin based on the followin	g configuration y
Detection Policy				
Detection Policy	connectivity,	HTTPS/HTTP Suitable for applications that need to	TCP Suitable for scenarios that require	UDP Suit for sco
Detection Policy ICMP Ping Only check network of host accessibility	connectivity,	HTTPS/HTTP Suitable for applications that need to recognize the content of the request, such as Web applications, app services, etc.	TCP Suitable for scenarios that require high reliability and data accuracy and low transmission speed, such as file transfer, remote login, etc.	UDP Suit for scc transmissi low accura messaging
Detection Policy ICMP Ping Only check network of host accessibility Basic configuration	connectivity,	HTTPS/HTTP Suitable for applications that need to recognize the content of the request, such as Web applications, app services, etc.	TCP Suitable for scenarios that require high reliability and data accuracy and low transmission speed, such as file transfer, remote login, etc.	UDP Suit for scc transmissi low accura messaging
Detection Policy ICMP Ping Only check network of host accessibility Basic configuration Detection frequency	connectivity, 1 Every 30 secon	HTTPS/HTTP Suitable for applications that need to recognize the content of the request, such as Web applications, app services, etc.	TCP Suitable for scenarios that require high reliability and data accuracy and low transmission speed, such as file transfer, remote login, etc.	UDP Suit for scc transmissie low accura messaging

Disable

③ Traffic schedulin

is that require high iciency and relatively ich as instant ne video, etc.

Disable all hea

g	1				
Detection frequency	Every 30 seco	nds		•	
Collapse Advanced Con	figuration 🔺				
Source station hea	th conditions				
Timeout period	_ 5	+ accordo			
rineout period	- 5	T Seconds			
	the default is 5 s	seconds.			
Unhealthy threshold	- 2	+ times			
	The number of t	imes to retry when the health check re	sult is "Unhealthy"		
		1 march			
L Le e lélecci éle con e le e l el	- 3	+ times			
Healthy threshold					

Configuration Item	Description
Timeout	The allowed timeout duration for a single health check request to the origin. If no response is received within this period, the origin is considered Unhealthy. The default is 5 seconds, with a configurable range of [1, 30] seconds.
Unhealthy Threshold	The number of consecutive probe failures required to determine an origin Unhealthy. Once this threshold is reached, the origin is considered Unhealthy. The default is 2 times, with a configurable range of [1, 5]. For example, if this value is set to 2, and an origin is initially Healthy, upon receiving two consecutive Unhealthy probe results, the origin will be considered Unhealthy.
Healthy Threshold	The number of consecutive successful probes required to restore an origin to a Healthy state, making it available again. The default is 3 times, with a configurable range of [1, 5]. For example, if this value is set to 3, and an origin is Unhealthy, after three consecutive Healthy probe results, the origin will be restored to a Healthy status.

Active Probing Cycle for Origin Health Status Change





Total duration: 1+30+1+30+1=63s

For example, suppose the health determination conditions for the origin are set as follows: timeout of 5 seconds,

unhealthy threshold of 3 times, healthy threshold of 3 times, and a probe interval of every 30 seconds.

The time required to consider an origin Unhealthy would then be: 5+30+5+30+5=75 seconds.

The time required to restore the origin to a Healthy state (assuming a successful active probe response takes 1 second) would be: 1+30+1+30+1=63 seconds.

More Information

Introduction to Principle of Probe Method

ICMP Ping HTTP/HTTPS TCP UDP



The ICMP Ping health check mechanism is as follows:

1. EdgeOne probe node sends a Ping command to your origin.

2. If the Ping is successful, and within the backsource timeout period, the origin receives an ICMP reply, the service is considered normal, and the result of this check is considered healthy.

3. If the Ping fails, and within the backsource timeout period, the probe node does not receive an ICMP reply from the origin, the service is considered abnormal, and the result of this check is considered unhealthy.

Note:

ICMP Ping requires your origin to support Ping.



The HTTP/HTTPS health check mechanism is as follows:

1. EdgeOne probe node sends an HTTP request to your origin. It requires configuration of the corresponding URL and port, with the option to include a custom HOST header.

2. If, within the backsource timeout period, the EO probe node receives an HTTP status code from the origin that successfully matches the configured HTTP status codes, the result of this check is considered healthy.

3. If, within the backsource timeout period, the EO probe node does not receive a response from the origin or receives a status code that does not match the configured codes, the result of this check is considered unhealthy.

S Tencent Cloud



The TCP health check mechanism is as follows:

1. EdgeOne probe node sends a SYN connection request packet to a specific port (configurable) on your origin.

2. Upon receiving the SYN request packet, if the corresponding port on the origin is in a normal listening state, it will respond with a SYN+ACK packet.

3. If, within the backsource timeout period, the probe node receives a SYN+ACK response packet from the origin, it indicates that the service is running normally. The result of this check is considered healthy. The probe node then replies with an ACK packet to the origin and sends an RST reset packet to terminate the TCP connection.

4. If, within the backsource timeout period, the probe node does not receive a SYN+ACK response packet from the origin, it indicates that the service is running abnormally. The result of this check is considered unhealthy. The probe node sends an RST reset packet to the origin to terminate the TCP connection.



The UDP health check mechanism is as follows:

1. EdgeOne probe node sends a customized probe packet to a specific port (configurable) on your origin.

2. If, within the backsource timeout period, the probe node receives a customized response packet from the origin, it indicates that the service is running normally. The result of this check is considered healthy.

3. If, within the backsource timeout period, the probe node does not receive a customized response packet from the origin or receives a response packet that does not conform to the defined content, it indicates that the service is running abnormally. The result of this check is considered unhealthy.

Note:

Both the request content and response content are customized, and you need to configure the corresponding requestresponse content on your origin.

Probe Request Identification

Active probes do not carry special request identifiers. When you choose ICMP Ping or TCP probes, there are no related features. When you choose UDP probes, customized content can be configured to serve as identifiers. For HTTP/HTTPS probes, separate customized request headers can be configured to serve as identifiers.

Viewing the Health Status of Origin Server

Last updated : 2024-05-29 10:33:37

The node probe results will display the outcomes of probes initiated by EdgeOne from various nodes and regions within the global availability zones towards the current origin group. Users can view these probe results to find whether the origin is healthy across different zones.

Note:

EdgeOne Load Balancing feature is in beta testing. If you want to use it, you can Contact Us.

1. Log in to the EdgeOne console. In the left menu bar, click the **Site List**. Within this list, click the **Site** need to be configured to go to the details page.

2. On the site details page, click **Origin Settings** > Load Balancing.

3. On the Load Balancing page, click the desired Load balancer.

Create Instance				Please enter the instance name	0
Load balancer/ID	Instance type	Health Check Policy	Traffic scheduling policy	Status	Operation
load_balancer_1	HTTP-specific type	ICMP Ping detection	Failover in order of priority	O Activated	Edit Delete
<i>1</i> 0	HTTP-specific type	ICMP Ping detection	Failover in order of priority	O Activated	Edit Delete
Total items: 2				10 💌 / pag	ge 🖂 🔺 1 /1 page

4. In the instance details page, click View details.

Instance details				
Instance name	load_balancer_1 HTTP-specific type ICMP Ping detection	n		
Instance ID	Г <u>а</u>			
Check frequency	Every 30 seconds			
The timeout(in seconds) before making the health check failed.	5 seconds			
Thresholds to mark origin unhealthy.	2 times			
Health threshold	3 times			
Traffic scheduling policy	Failover in order of priority			
Request retry policy	When an origin is marked unhealthy or request an origin fails,	, subsequent requests go directly to the next prioritized group of origin.		
Origin Group Status	Priority Origin Group	Origin Health Status	Origin group type	Operation
	 March 	ipv4	HTTP-specific type	View details

5. In the node probe results, nodes are differentiated by the following three colors:

Green Node: Indicates that the probe node in the region has considered all origins in the origin group to be healthy. **Red Node:** Indicates that the probe node in the region has considered one or more origins in the origin group to be unhealthy.

🔗 Tencent Cloud

Gray Node: Indicates that the probe node in the region cannot probe any origins. Probing is done at the IP level, meaning if the origin is a domain, the domain will be resolved into an IP before it probes. This situation usually occurs if you have entered an incorrect domain origin which cannot be resolved into an IP. In this case, it is recommended to check for potential spelling mistakes in the origin domain or whether the corresponding domain has expired.



Note:

Probe nodes in different regions make independent decisions. Edge nodes will route requests back to the origin based on the probe results from the nearest probe nodes in each region.

For example: If your origin is in Hong Kong (China), and the probe node in Singapore considers the origin to be unhealthy whereas the probe node in Germany considers it to be healthy, traffic from the Singapore region will not be routed to that origin, while traffic from the Germany region will continue to be directed to that origin.

In the scenario described above, you can refer to the probe results from other regions for a comprehensive view. If only a few nodes consider the origin to be unhealthy, it might be due to network fluctuations in certain areas. If the majority of nodes consider the origin to be unhealthy, it is recommended to check whether the origin has malfunctioned.



Related References Load Balancing-Related Concepts

Last updated : 2024-05-29 10:33:37

This document introduces the relevant concepts involved in Load Balancing.

Note:

Tencent Cloud EdgeOne Load Balancing feature is in beta testing. If you want to use it, you can Contact Us.

Load Balancer

A load balancer is a virtual concept, comprising Origin groups and health check policies. Within a single load balancer, up to ten origin groups can be configured in priority order, accompanied by one health check policy. The load balancer intelligently directs business traffic based on probe results and the configured traffic scheduling policy.



Origin Group

An origin group is the smallest unit of origin configuration within the Load Balancing. You can add one or more origins. When you add multiple Origin Servers, you can configure weights to adjust traffic load. For more details, see Origin Group Configuration.

Health Check Policy

The health check policy consists of probe methods and health assessment criteria. Currently, four probe methods are supported: ICMP Ping, HTTP/HTTPS, TCP, and UDP. For more details, see <u>Detailed Health Checks</u>.

Traffic Scheduling Policy

The traffic scheduling policy only takes effect when the health check policy is enabled. Currently, it supports a Failoverby-Priority-Order policy, that is, based on probe results, it disables failed origins and routes traffic to healthy ones according to the priority order of origin groups.

Request Retry Policy

In the event of a request failure to a particular origin during normal business operations, the Load Balancing feature, guided by its request retry policy, can schedule the request to another origin for a retry. This helps reduce business request failures due to network issues or origin malfunctions. For more details, see the Introduction to Request Retry Policy.

Introduction to Request Retry Strategy

Last updated : 2024-05-29 10:33:37

Load Balancing is capable of redirecting a request to an alternative origin for retrial when a request to an initially designated server fails, in accordance with the retry policy. This reduces business request failures caused by network issues or origin fails.

Note:

EdgeOne Load Balancing feature is in beta testing. If you want to use it, you can Contact Us.

Actual business requests may fail due to the following reasons:

1. Origin Failure and Have Not Actively Probe to Disable: After health check policies are configured, active probe is conducted periodically. Traffic is directed based on the results of the previous probe until new results are available. If an origin becomes unhealthy between two probes, business traffic might still be directed to the unhealthy origin. This leads to business request failure.



2. **Network Jitter:** The origin is healthy, but network issues occur during the access. This leads to business request failure.

Note:

Request failures include origin connection establishment failures and origin response reception failures.

For the situations mentioned above, EdgeOne provides the following two fallback request retry policies:

Policy 1: When a real business request fails to access a certain origin, it directly retries with another origin within the next lower priority origin group. This is suitable for scenarios where the performance of both higher and lower priority origin groups is similar.

Policy 2: When a real business request fails to access a certain origin, it directly retries with another origin within the same priority origin group. This is suitable for scenarios where the performance of the higher priority origin group is



significantly better than that of the lower priority origin group.

Origin Group Configuration

Last updated : 2024-08-01 21:32:16

Overview

Manage business origins in the form of origin groups. The origin groups configured here can be used in functions such as adding acceleration domain names and L4 proxy.

Create Origin Group

1. Log in to the EdgeOne console and click Site List in the left sidebar. In the **site list**, click the target site to enter the site details page.

2. On the site details page, click origin configuration > origin group.

- 3. Click Create origin group.
- 4. Fill in the origin group name and select the origin type. The specific type descriptions are as follows:

HTTP Dedicated: Supports adding IP/domain name origins and object storage origins, and can only be used

for site acceleration-related services (e.g., Domain Name Service and rule engine - Modify origin).

Universal: Only supports adding **IP**/**domain name** as origin, does not support adding **object storage origin**, and can be used for site acceleration services (such as Domain Name Service and rule engine) and L4 proxy.

Note:

After the configuration is complete, the origin group type cannot be modified.

Create origin group				
Origin group name				
	1-200 characters ([a-z], [A-Z], [0-9], [_])		
Origin group type	O HTTP-specific type Genera	ıl Туре		
	HTTP-specific origin groups support	"IP/Domain" and "Object Storage Bucket" as origin, but can only be refere	enced by the Layer 7 acceleration services (Domain Service a	and Rule Engine).
Origin server	Origin type	Origin address	Weight 🕄	Operat
	+ Add origin			
Host Header(optional)	Please enter origin Host Header.			
	If your origin-pull host is different from Note: If you configure the object store At the same time, the rule engine mod	n the accelerated domain name, you can use this feature to rewrite the hos age origin, this configuration does not modify the host to ensure that the or dification of the host-related operations has a higher priority.	st to the actual host. rigin request will not fail.	
Create	icel			



5. Click the **Add Origin** button to configure the origin. The supported origin types are as follows, with up to 20 origins supported.

Object storage origin: Tencent Cloud COS or other object storage buckets compatible with AWS S3.

IP/domain name origin: Supports IPv4 addresses, IPv6 addresses, and domain names as origins.

Note:

Explanation of weight-related configurations in the origin group:

1. If a weight is set for an origin in the origin group, all origins in the group must also set corresponding weights. Weights can be integers between 0 and 100. If the weight of an origin is set to 0, no origin-pull requests will be allocated to that origin. Other non-zero weight origins will be allocated origin-pull requests based on their respective weight ratios.

2. If you do not set a weight, all origins in the origin group should not set weights at the same time. In this case, if "smart acceleration" is not enabled, EdgeOne will distribute origin-pull requests equally to each origin. If "smart acceleration" is enabled, EdgeOne will select the best quality origin for each origin-pull request.

Create origin	
Origin type	IP/Domain Origin 🔹
Origin (IP/Domain name)	Please enter IPv4/IPv6/domain or
Weight (optional)	
	Any integer from 0-100 is supported.
	Create Cancel

6. Click **Create** to complete the origin group creation.

Origin-pull configuration Origin-Pull Timeout

Last updated : 2024-09-04 17:53:02

Overview

The EdgeOne rule engine supports setting custom origin-pull timeouts. You can reasonably set the origin-pull request timeout based on the network link conditions and the data processing capability of the origin server, to ensure normal origin-pull for the request. The origin-pull timeout is defined as follows. If there is no data response from the origin server after a node initiates an origin-pull request, no matter how long the duration is, the node will consider it a timeout and actively disconnect from the origin server.

Currently, it supports configuring the HTTP response timeout (please stay tuned to the support for TCP connection timeout configuration, which is coming soon). The timeout can be set to an integer from 5 to 600, with a default value of 15. It means that after a node is connected to an origin server and initiates an HTTP request, if the origin server does not respond with any data within 15 seconds (including scenarios where there is no data response at all or partial data response is interrupted), the node will consider it an HTTP response timeout.





Scenario: Configuring the HTTP Response Timeout to 60 Seconds

If the origin server for your business at the www.example.com domain name under the example.com site is heavily loaded and takes longer processing time, you should extend the timeout to 60 seconds, to avoid access failure due to active disconnection of the node after the default HTTP timeout of 15 seconds. You can refer to the following steps:

1. Log in to the EdgeOne console, click Site List in the left sidebar, and then click the site you want to configure in the site list.

2. On the site details page, click **Site Acceleration** to enter the global configuration page. Then click the **Rule Engine** tab.

3. On the rule engine page, click Create rule and select Add blank rule.

4. On the rule editing page, select the matching type as HOST and set its value to equal www.example.com .

5. Click the **choice box** below **Action** and select the action as **Upstream Timeout** in the pop-up action list. Then configure the HTTP response timeout to 60 seconds.

6. The complete rule configuration is shown below. Click **Save and publish** to finish the rule configuration.

IF + Comment			
Matching type ①	Operator	Value	
HOST .	ls 💌		
+ And + Or			
Action ①	HTTP Response Timeout ③		
Upstream Timeout	- 60 + se	econds	
+ Action			
+ IF			

Configuring Origin-Pull HTTPS

Last updated : 2024-08-26 10:54:37

Overview

You can specify the protocol that EdgeOne uses in the origin-pull request.

In scenarios that requires a high level of security, HTTPS can be used to access a website to ensure the security of website data. When HTTPS is specified as the origin-pull protocol, all origin-pull requests from EdgeOne to the origin use HTTPS, which prevents data tampering or theft during transmission.

In scenarios where fast response is required, HTTP can be used for origin-pull requests to speed up website access. When HTTP is specified as the origin-pull protocol, you can avoid complex SSL handshakes and other operations between EdgeOne and the origin, thus improving the website access speed. If your origin does not support HTTPS, please select HTTP.



1. An EdgeOne node initiates an origin-pull request by using the specified origin-pull protocol.

2. The origin responds to the request and establishes a connection by using the same protocol as the request.

Note:

The configuration priority of the rule engine is superior. If the origin protocol rule is configured simultaneously within the domain name service and the rule engine, the final standard is determined by the rule engine.

Scenario 1: Configuring origin-pull HTTPS for multiple domain names in batches in the rule engine

If you need to uniformly change the origin-pull protocol to origin-pull HTTPS for multiple domain names, such as

www.example.com , vod.example.com and image.example.com , please refer to the following steps:

1. Log in to the EdgeOne console and click Site List in the left sidebar. In the site list, click the target Site.

2. On the site details page, click **Site Acceleration** to enter the global site configuration page, then click the **Rule Engine** tab.

3. On the rule engine management page, click Create rule and select Add blank rule.



4. On the rule editing page, enter the rule name and select Host matching type to match the request of the specified

domain name. In the current scenario, select domain names www.example.com , vod.example.com and

image.example.com .

5. Click on Action > Select Box, select Origin-pull HTTPS from the dropdown action list that appears.

+ Comment				
Matching type ④	Operator	Value		
HOST		· · · · · · · · · · · · · · · · · · ·		
+ And + Or				
Action ①	Protocol			
Origin-pull HTTPS	HTTPS	*		
+ Action				
+ IF				

6. Click on Save and Publish to finalize this rule configuration.

Scenario 2: Configuring origin-pull HTTPS for the specified domain name

If you need to specify a domain name to modify the origin-pull protocol into origin-pull HTTPS, such as

www.example.com, please follow these steps:

- 1. Log in to the EdgeOne console and click **Site List** in the left sidebar. In the site list, click the target **Site**.
- 2. Choose Domain Name Service > Domain Name Management on the Site Details page.
- 3. Select the domain name that needs to be modified and click Edit on the Domain Management page.

		Object storage ori	gan - se ann ge	Activated	Not configured	Edit Switch to On
Domain name	Extended service	Origin type	Origin settings	Status	HTTPS certificate	Operation

4. In the origin-pull protocol, select HTTPS and click Complete to finish the modification.



Edit domain name	
Domain name	
Origin type	O IP/Domain name Object storage origin Origin Group
Origin (IP/Domain name)	
IPv6 access	O Follow site configuration: Disable Enable Disable
Origin Protocol	Follow protocol HTTP HTTPS
Origin Port	HTTP 80 HTTPS 443

Origin configuration guid

Follow protocol

Client requests with HTTP or H protocol, EdgeOne follows the client's protocol to request the (the origin needs to support bot 80 and port 443, otherwise it m to request origin)

HTTP

Use HTTP protocol request the origin, the default port 80, supp the configuration of custom por

HTTPS

HTTPS protocol request the original the default use of port 443, sup the configuration of custom por

Host Header Rewrite

Last updated : 2024-10-28 15:34:17

Overview

Host header rewriting enables you to rewrite the host header as the actual origin domain name when the origin domain name is different from the acceleration domain name in the origin group list.

Directions

1. Log in to EdgeOne console and click Site List in the left sidebar. In the site list, click the target site.

2. On the site details page, click **Site Acceleration** to enter the global site configuration page. Then click the **Rule Engine** tab.

3. On the rule engine management page, click Create rule and select Add blank rule.

4. On the rule editing page, set the matching type as HOST and configure it as the domain name to be modified, such

as www.example.com .

5. Click Action > choice box and select the action as **Rewrite host header** in the dropdown action list. You can select the mode as **Custom** or **Follow origin domain**.



6. Click **Save and publish** to complete the rule configuration.

Controlling Origin-pull Requests

Last updated : 2024-08-26 10:54:37

Overview

By default, when origin-pulling, all query strings and Cookies within the request will be retained. If your business origin only allows specified query strings or Cookie information to be carried in the origin-pull request, you can ensure the normal origin-pull request by deleting the specified origin-pull request parameters.

Directions

For example, Client requests Request URL: http://www.example.com/path/demo.jpg?

key1=a&key2=b&key3=c&key4=d, and only key1=a parameter needs to be retained when origin-pulling. You can follow the steps below to configure:

1. Log in to the EdgeOne console and click **Site List** in the left sidebar. In the site list, click the target **Site**.

2. On the site details page, click **Site Acceleration** to enter the global site configuration page, then click the **Rule Engine** tab.

3. On the rule engine management page, click Create rule and select Add blank rule.

4. On the rule edit page, select the matching type as HOST equals www.example.com.

5. Click on the **action**, and in the pop-up operation list, select the operation as **origin-pull request parameter settings**.

 Select the mode as retaining specified parameters, Enter the parameters key1 and key2 to be retained, up to 10 parameters are allowed.

IF	+ Comment		
	Matching type ①	Operator	Value
	HOST v	ls 💌	
	+ And + Or		
	Action ①		
	Origin-pull request par		
	Туре	Mode	Parameter O
	Query string	Reserve Specified Para 💌	key1;key2
	+ Add		
1	► Action		
	+ IF		

7. Click Save and Publish to complete the rule Configuration.

Redirect Following During Origin-Pull

Last updated : 2024-08-26 10:54:37

Overview

Under normal circumstances, when the origin returns a 301/302 request, the node will return the status code to the client by default, and the client will redirect to the corresponding resources for access. EdgeOne supports follow origin redirects. When enabled, if the node receives a 301/302 status code during origin-pull, it will actively follow the redirect (not exceeding the set maximum redirects) to the specified address until the corresponding file is obtained, and then respond to the client with the actual resources, which can improve the user's access response speed.

For example: The client accesses the URL https://a.example.com/test.jpg, the origin A redirects the URL 302 to https://b.example.com/test.jpg, and the domain a.example.com has accessed the EdgeOne Service, while b.example.com has not yet accessed the acceleration service. Then:

Without enabling origin-pull follow redirect: After the client initiates the visit, if there is no cache in the EdgeOne node, it will visit the origin A and receive the 302 status code, and then respond to the client with the status code, and the client will directly request the origin B for the corresponding resources. At this time, since the origin B has not accessed the acceleration service, the client's self-initiated access speed is slower, and the obtained file cannot be cached. When other users access the same file, the process needs to be repeated.

Enable origin-pull follow redirect: After the client initiates the visit, if there is no cache in the EdgeOne node, it will visit the origin A and receive the 302 status code, and then, according to the status code and the corresponding address, directly request the origin B for the corresponding resources, and cache the resources in the node. This process is carried out by the EdgeOne node for origin-pull requests, the request speed is faster, and the obtained file can be cached in the node. When other users access the same file, there is no need to repeat the origin-pull, and the file can be directly hit and responded to the client.



Directions

For example: If you need to enable origin-pull follow redirect for the specified domain www.example.com, with a
maximum of 3 redirects. You can refer to the following steps:

1. Log in to the EdgeOne console and click Site List in the left sidebar. In the site list, click the target Site.

2. On the site details page, click **Site Acceleration** to enter the global site configuration page, then click the **Rule Engine** tab.

3. On the rule engine management page, click Create rule and select Add blank rule.

4. On the rule editing page, select the matching type as HOST equals www.example.com .

5. Click on the **Action**, and in the pop-up operation list, select the operation as **follow origin redirect**.

6. Click on the switch, click on the switch to enable, and set the maximum redirects to 3 times. The related configuration instructions are as follows:

Maximum redirects: You can set it between 1-5 times. Within the maximum redirects, the node will follow the redirect address until the corresponding resources are obtained. If the maximum redirects are exceeded, the corresponding status code will be directly responded to the client.

+ Comment					
Matching type ③	Operator	Value			
HOST	• Is		8		
+ And + Or					
Action ①	On/Off Maximum	n redirects			
Follow origin redirect	—	3 +			
Action					

7. Click on **Save and Publish** to complete the rule configuration.

HTTP/2 Origin-Pull

Last updated : 2024-08-26 16:03:35

Overview

Support EdgeOne nodes to origin-pull using HTTP/2 protocol. HTTP/2 (i.e., HTTP 2.0, Hypertext Transfer Protocol version 2) is the second major version of the HTTP protocol, which can effectively reduce network latency and improve site page loading speed.

Note:

- 1. When enabled, the origin must support HTTP/2 protocol access.
- 2. If you need to configure HTTP/2 access, please refer to HTTP/2.

Use Limits

When HTTP/2 origin-pull is enabled and the origin-pull protocol is set to Follow Protocol, if a client sends an HTTP request, the EdgeOne node will perform origin-pull using H2C. However, if the origin server does not support H2C, the origin-pull will fail.

Therefore, if your current origin server does not support H2C and the origin-pull protocol is set to Follow Protocol, to reduce the risk of origin-pull failure, we recommend keeping HTTP/2 origin-pull disabled for the site/domain name. If your origin-pull protocol is HTTPS, it will not be affected.

Note:

H2C is an unencrypted version of HTTP/2, where C stands for clear text, namely plaintext. HTTP/2 is the second major version of the HTTP protocol, with significant performance improvements including multiplexing of requests and responses, reduced latency, optimized data flow, and header compression. However, the HTTP/2 protocol is typically used over secure HTTPS, requiring encryption by TLS (Transport Layer Security Protocol), while H2C allows the use of HTTP/2 without encryption, enabling performance benefits of HTTP/2 when encryption is not needed or cannot be used. Therefore, EdgeOne uses H2C for origin-pull when HTTP/2 origin-pull is enabled and HTTP is used for origin-pull.

Directions

If you need to enable or disable HTTP/2 origin-pull for the specified domain www.example.com , you can follow
the steps below:



1. Log in to the EdgeOne console, click Site List in the left sidebar, and then click the site you want to configure in the site list.

2. On the site details page, click **Site Acceleration** to enter the global configuration page. Then click the **Rule Engine** tab.

3. On the rule engine page, click **Create rule** and select **Add blank rule**.

4. On the rule editing page, select the matching type as HOST equals www.example.com .

5. Click the **Action**, and in the pop-up operation list, select the operation as **HTTP/2 origin-pull**. Click the switch to enable/disable HTTP/2 origin-pull.

+ Comment			
Matching type ①	Operator	Value	
HOST	▼ Is	· · · · · · · · · · · · · · · · · · ·	
+ And + Or			
Action ①	On/Off		
HTTP/2 origin-pull			
Action			
15			

6. Click Save and Publish to complete the rule configuration.

Range GETs

Last updated : 2024-08-26 15:57:58

Overview

Range GETs can be enabled to reduce the consumption of large file origin-pulls and response time.

Why can Range GETs improve the efficiency of large file delivery?

When caching large files, nodes will split them into smaller parts in order to improve cache efficiency. All parts cached expire at the same time and follow the node cache TTL configuration. Range requests are also supported. For example, if a client request carries the HTTP header Range: bytes = 0-999, only the first 1000 bytes of the file will be returned to the user.

If Range GETs is enabled: When parts of the file are requested and their caches expire, nodes only pull and cache the requested parts and return them to the user, so that origin-pull consumption and response time are greatly reduced. If Range GETs is disabled, when the client requests only parts of a file, the node will pull only the requested parts according to the Range header in the client request, cache them, and return them to the client at the same time. However, this may not be able to achieve the optimal performance. In large file scenarios, we recommend you enable Range GETs.

Use Cases

You can use Range GETs to cache large static files in either of the following cases: The origin server supports Range requests, or you use a Tencent Cloud COS origin server and do not apply any data processing methods such as image processing.

Notes

The origin server must support Range requests, or the origin-pull may fail.

The origin-pull may fail if Range GETs is enabled for small static files, or if you enable it while using a Tencent Cloud COS origin server and data processing methods such as image processing.

Directions



For instance, you have a video service website that provides online video watching through

video.example.com . The videos are mainly long videos with large files. In order to reduce traffic consumption of large files and improve origin-pull speed, you need to support range requests and origin-pull. You can perform the following steps:

1. Log in to the EdgeOne console and click **Site List** in the left sidebar. In the site list, click the target **Site**.

2. On the site details page, click **Site Acceleration** to enter the global site configuration page, then click the **Rule**

Engine tab.

3. On the rule engine management page, click Create rule and select Add blank rule.

- 4. On the rule editing page, select the Matching type as HOST equals video.example.com .
- 5. Click on Action, in the displayed operation list, choose the operation as Range GETs.
- 6. Click on **On/Off** to enable Range GETs.

HOST value HOST value + And + Or Action O On/Off	HOST v Is + And + Or Action © On/Off	
+ And + Or Action O On/Off	+ And + Or Action © On/Off	
Action 0 On/Off	Action () On/Off	
	Action O On/On	
	Range GETs	

7. Click on **Save and publish** to complete the configuration of this rule.

Related References Id Version Origin Group Compatible Related Issues

Last updated : 2023-10-24 15:45:49

The origin group has carried out a product capability upgrade since October 24, 2023. After the upgrade, the old version of the origin group will be processed for compatibility in the following ways. At the same time, we also suggest you switch to the usage of the new version of the origin group.

Origin type & Configuration method compatibility

The new version of the origin group will no longer distinguish between **self-owned origin**, **object storage origin**, and **Tencent Cloud COS type origin**. The original origin groups with origin type of **object storage and Tencent Cloud COS** will be automatically updated to the new version of dedicated **HTTP origin group**, and the original origin groups with **self-owned origin** type will be automatically updated to the **universal origin** group. The origin group will no longer support the configuration of origin-pull by region/protocol. If you have previously configured related origin-pull rules by region/protocol, the rules will be migrated to the rule engine as shown below:

modify origin-http/https 🖉			
IF			
HOST Is			
IF			
Request protocol Is HTTP			
Modify origin Origin type: Origin Group	Origin Group: Origin Prote	ocol: HTTP Port: 80	
IF			
Request protocol Is HTTPS			
Modify origin Origin type: Origin Group	Origin Group: Origin Prote	ocol: HTTPS HTTPS origin port: 4	43



modify origin-region 🖉											
IF											
но	ST Is	<u>n</u>									
IF											
	Client geo loca	ation Is Asia 🛅									
	Modify origin	Origin type: Origin Group	Origin Group:	Origin Protocol: Follow protocol	Port: 80	HTTPS origin port: 443					
IF	÷.,										
	Client geo loca	ation Is Europe 🗇									
	Modify origin	Origin type: Origin Group	Origin Group:	Origin Protocol: Follow protocol	Port: 80	HTTPS origin port: 443					

Origin group port migration description

The new version of the origin group will no longer support port configuration. All port configurations will be migrated to the service configuration entry, such as L4 proxy or Domain Management.

Domain configura	ition	>	2 Re co on	ecommended nfiguration(O al)	> (3 Configure	CNAME	
Oomain name							Domain Configuration	
Origin type Origin Group	O IP/Do	Object storage origin Select from existing origin groups			Origin Group	C Load balancing	IP/Domain name It can be an IPv4/IPv6 address or a domain name.	
Origin Protocol Origin Port	Follow protocol	80	HTTP H	 HTTPS 443 			 Object storage origin The object storage source site of cloud storage service providers, currently supports storage buckets of Tencent Cloud COS and Amazon AWS Signature V4 protocols Origin Group Applicable to a single domain name back to the origin of multiple origin station, multiple domain names share the same origin station configuration. Load balancing Proactively detects the delay and health status of the origin, configures intelligent traffic scheduling policies, and provides safer and faster traffic distribution services. 	ts of in hare ion. ures es, fic
					Cancel	Next		
	-	Forw	arding	Origin type	Ovisin odd		Session persistence	

Primary and Standby Origin Configuration Instructions

Origin Gr 👻

100-110

In the **Domain Management** and **Rule Engine - Modify Origin**, directly configuring primary and standby origins is no longer supported. Existing configurations will not be affected, but modifications are no longer supported. If you currently have a demand for primary and standby origin configurations, please contact us for support.

test

100-110

Ŧ

TCP 🔻

Т