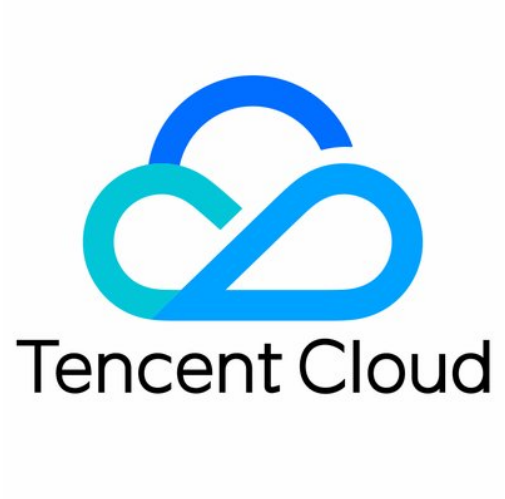
Tencent Cloud EdgeOne

Practical Tutorial

Product Documentation



Contents

Practical Tutorial
Automatic Warm-up/Cache Purge
EdgeOne + COS Realizes Automatic Warm-up

Last updated：2025-06-23 15:01:18

This document provides an overview of how to achieve EdgeOne automatic pre-warming resources with Tencent Cloud Object Storage (COS) and Serverless Cloud Function (SCF) through EdgeOne. For details on pre-warming functions and principles, see URL Pre-Warming.

Background Introduction

If your origin server is Tencent Cloud Object Storage (COS), when new hot resources are uploaded to the origin server (such as APK installation packages, popular videos, course files, etc.), it is usually necessary to pre-cache the resources to EdgeOne edge nodes through cache pre-warming. This is to avoid situations where, upon the client's initial request, the resources are not cached at the node, leading to a request being sent back to the origin server. However, manual submission of URLs that need pre-warming in the EdgeOne console after uploading files to Tencent Cloud COS is required. In cases with many URLs for pre-warming, this process can be prone to omissions and delays due to manual operations.

Automatic pre-warming can assist you in detecting and invoking EdgeOne's cache pre-warming API through Tencent Cloud Serverless Cloud Function (SCF) after uploading files to Tencent Cloud Object Storage (COS). This process ensures that your files are pre-warmed to EdgeOne nodes immediately after upload, enhancing cache hit rates and reducing the number of origin-pull requests.

Note:

Tencent Cloud Object Storage (COS) is a paid feature, and charges incurred during usage are collected by Tencent Cloud COS. For specific charging details, see COS Billing Overview.

Serverless Cloud Function (SCF) is a paid feature, and charges incurred during usage are collected by Serverless Cloud Function (SCF). For specific charging details, see SCF Billing Overview.

There are daily limits on the number of pre-warms, with different limits for different billing plans. See Comparison of EdgeOne Plans for details.

Applicable Scenarios

Scenario 1: Releasing New Content
After uploading a new version of an installation package or upgrade package to Tencent Cloud COS, resources are automatically pre-warmed to EdgeOne acceleration nodes. Once the file is officially released, download requests from a massive number of users will be directly responded to by the acceleration nodes, improving download speeds and significantly reducing the load on the origin server.
Scenario 2: Large-scale Marketing Campaigns
Before the marketing campaign is launched, static resources related to the campaign page are uploaded to Tencent Cloud COS in advance. Resources are automatically pre-warmed to EdgeOne acceleration nodes. Once the campaign starts, users' access to static resources is responded to by acceleration nodes, reducing delays and congestion caused by high traffic.

Directions

Example Scenario

Assuming you are a game developer who has connected the site domain `www.example.com` to EdgeOne acceleration, and the source is Tencent Cloud COS with the address: `prefetch-cos-1251558888.cos.ap-guangzhou.myqcloud.com` .Because there are multiple game APKs that need frequent updates, you want the resources to be automatically pre-warmed to EdgeOne edge nodes immediately after uploading the APK.

Preparation

1. Ensure that COS and SCF services are activated, and record the bucket name and region information.
2. Follow the Quick Start to add your site, purchase the EdgeOne package, and obtain the site ID. The site ID can be found and copied from the site list after site access, for example, zone-2p42mkcpwz0y.

3. The acceleration domain name `www.example.com` has been added in the EdgeOne console, with the source configuration set to Tencent Cloud COS.

Step 1: Create and Deploy the Cloud Function for EdgeOne Automatic Pre-warming

1. Log in to the Serverless Cloud Function Console, and click on **Function Service** in the left-side menu bar.
2. On the Function Service page, click on **Create**, select **Template**, enter **EdgeOneAutomaticallyPrefetch** in the fuzzy search bar, select it, and click on **Next**.

3. On the "Function Configuration" page, the configurations below are required, and it is recommended to keep the other settings as default.

Basic Configuration

Function name: A function name will be automatically generated during function creation. You can choose to customize it for easy recognition.

Region: Select the region where the COS bucket is located, for example, Guangzhou.

Description: Explain the purpose of this function, such as using COS as a trigger. For example, when a file is uploaded to COS, it triggers the cloud function to complete the EdgeOne automatic pre-warming of files to the edge nodes.

Execution Role: Default selection is enabled. Configure and use the SCF template execution role. If using an existing role, ensure that the role includes the preset policies QcloudCOSFullAccess and QcloudTEOFullAccess.

Function Codes: The template already includes default function code implementing the EdgeOne automatic pre-warming capability. No modifications are necessary.

Environment Configuration

Click on **Advanced Configuration**, select **Environment Configuration**, and add the following key-value pairs to the environment variables. Keep the other configurations as default:

ZoneId: Fill in the ZoneId of the domain site `example.com` that needs automatic pre-warming. See the Preparation for obtaining the Site ID.

eoDomains: Fill in the accelerated domain names already added under ZoneId, such as `www.example.com` .

Note:

If you have multiple domain names in the current site using the same COS bucket as the source station and you want multiple domain names to trigger automatic pre-warming, you can add multiple environment variables starting with eoDomains, for example, eoDomains1, eoDomains2, as shown below.

Trigger Configuration

In the trigger configuration, select a COS Bucket that is in the same region as that of this SCF function. You can enter the bucket name for a fuzzy query, for example, `prefetch-cos-1251558888.cos.ap-guangzhou.myqcloud.com.` . Keep the other configuration items as default.

4. Click **Complete** to complete the creation of the EdgeOne automatic pre-warming function.

Step 2: Verification

1. Log in to the COS Console. In the left menu, click on **Bucket List**.
2. On the bucket list page, click on the **Bucket Name** used to **store the APK parent package**.

3. In the file list page, enter the root directory `prefetch-cos-1251558888.cos.ap-guangzhou.myqcloud.com` .
4. Click **Upload Files** and upload a file for the first time, for example, v2_src.apk, and then click **Upload**.

5. After successful file upload, in the SCF Console, click on the **Function Name** created in Step 1.
6. On the function management page, select **Trigger Management** > **Log Query** > **Invocation Logs**. Check the logs for successful invocation and ensure that the key information in the logs matches the uploaded file name, indicating successful triggering of the EdgeOne cache pre-warming API by SCF.

7. Go to the EdgeOne Console, enter the current site `example.com` , and click on **Site Acceleration** > **Cache Prefetching**.
8. On the cache pre-warming page, click on **History** to check if the pre-warming was successful. If it shows 'Success'', it indicates that the pre-warming has been completed.

9. Open developer tools in the browser and enter the file's access path, for example, `www.example.com/v2_src.apk` . Check the EO-Cache-Status value in the response header. If resources were not pre-warmed, the first access will show MISS. If it shows HIT, it means the resource has been automatically pre-warmed to the edge node, achieving cache hits even on the first access.

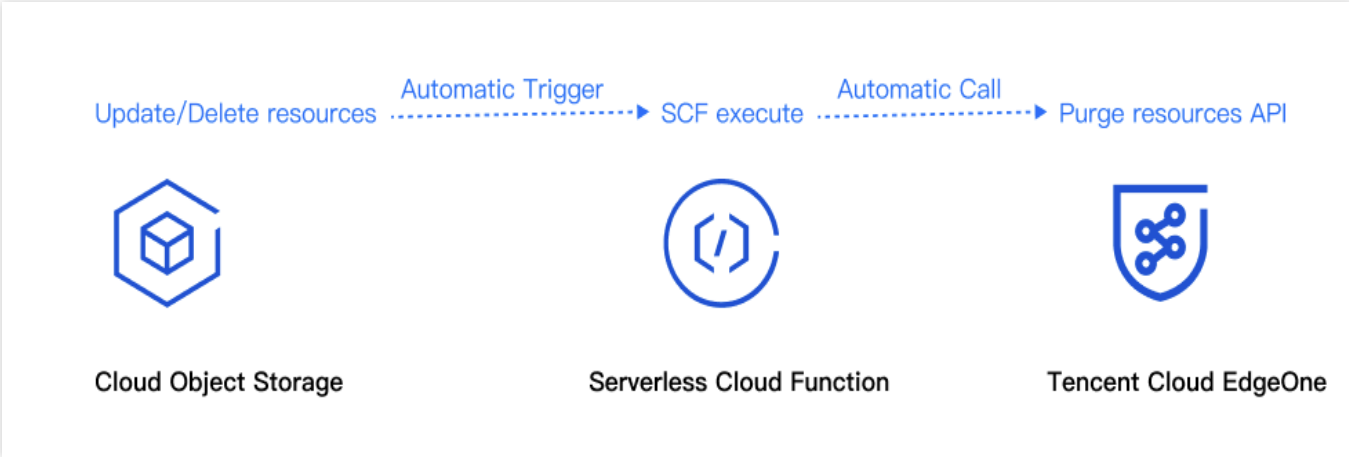
Monitoring Alarm (Recommendation)

EdgeOne sets quota limits for URL Pre-Warming tasks. For specific quotas, refer to Comparison of EdgeOne Plans. For URL Pre-Warming tasks that exceed the quota limit, an error will be triggered when calling the URL Pre-Warming API, and the final execution status of the SCF will be "calling failure". To promptly address this issue, it is recommended to configure SCF monitoring alarms through Cloud Monitor. For configuration methods, refer to: Configuring SCF Alarms, SCF Metric Descriptions.

EdgeOne + COS Realizes Automatic Cache Purge

Last updated：2025-01-24 16:33:19

This document mainly describes how to automatically purge the cache of EdgeOne by using Tencent Cloud Cloud Object Storage (COS) and Serverless Cloud Function (SCF). For the feature and principle of cache purge, refer to Cache Purge.



Background

If your origin server is a Tencent Cloud COS server, when there is an update of files with the same name or non-compliant resources requiring deletion on the origin server, it is usually necessary to also delete the resources from the EdgeOne node to avoid users still accessing old resources or non-compliant contents. However, cache purging requires you to manually go to the EdgeOne console or call the API interface to submit the URL requiring cache purge after you update or delete files on COS. This method can easily lead to omissions and may delay the completion of the purge due to manual operation.

After files are uploaded to Tencent Cloud COS, automatic cache purge can use Tencent Cloud SCF to help you automatically detect and call EdgeOne's cache purge API interface to automatically purge the node cache. This ensures that after your files are updated or deleted, users can immediately access the latest resources, thus enhancing user experience.

Note:

Tencent Cloud COS is a paid feature. Any possible fee incurred in use will be charged by Tencent Cloud COS. For specific charging details, refer to Overview of COS Billing.

SCF is a paid feature. Any possible fee incurred in use will be charged by SCF. For specific charging details, refer to Overview of SCF Billing.

Scenarios

Scenario 1: Updating a file with the same name

After a file is uploaded to Tencent Cloud COS and the file content is updated, re-uploading a file with the same name to COS will immediately make the latest resources accessible to users even before the CDN cache expires.

Scenario 2: Deleting a non-compliant file

A file uploaded to COS may certain a non-compliant content and need to be deleted from the COS bucket so that users cannot access the resource before the CDN cache expires.

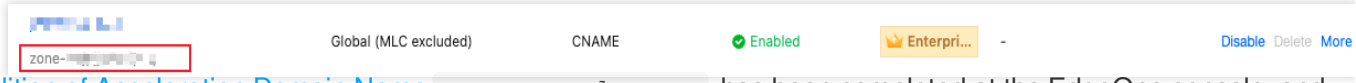
Directions

Sample Scenario

Assume you are a game developer and have connected the site domain name `www.example.com` to EdgeOne Acceleration, with the origin server being Tencent Cloud COS and located at: `purge-cos-1251558888.cos.ap-guangzhou.myqcloud.com` . Since multiple game APKs need frequent updates, you want the cache on the EdgeOne node to be automatically purged when files change.

Preparations

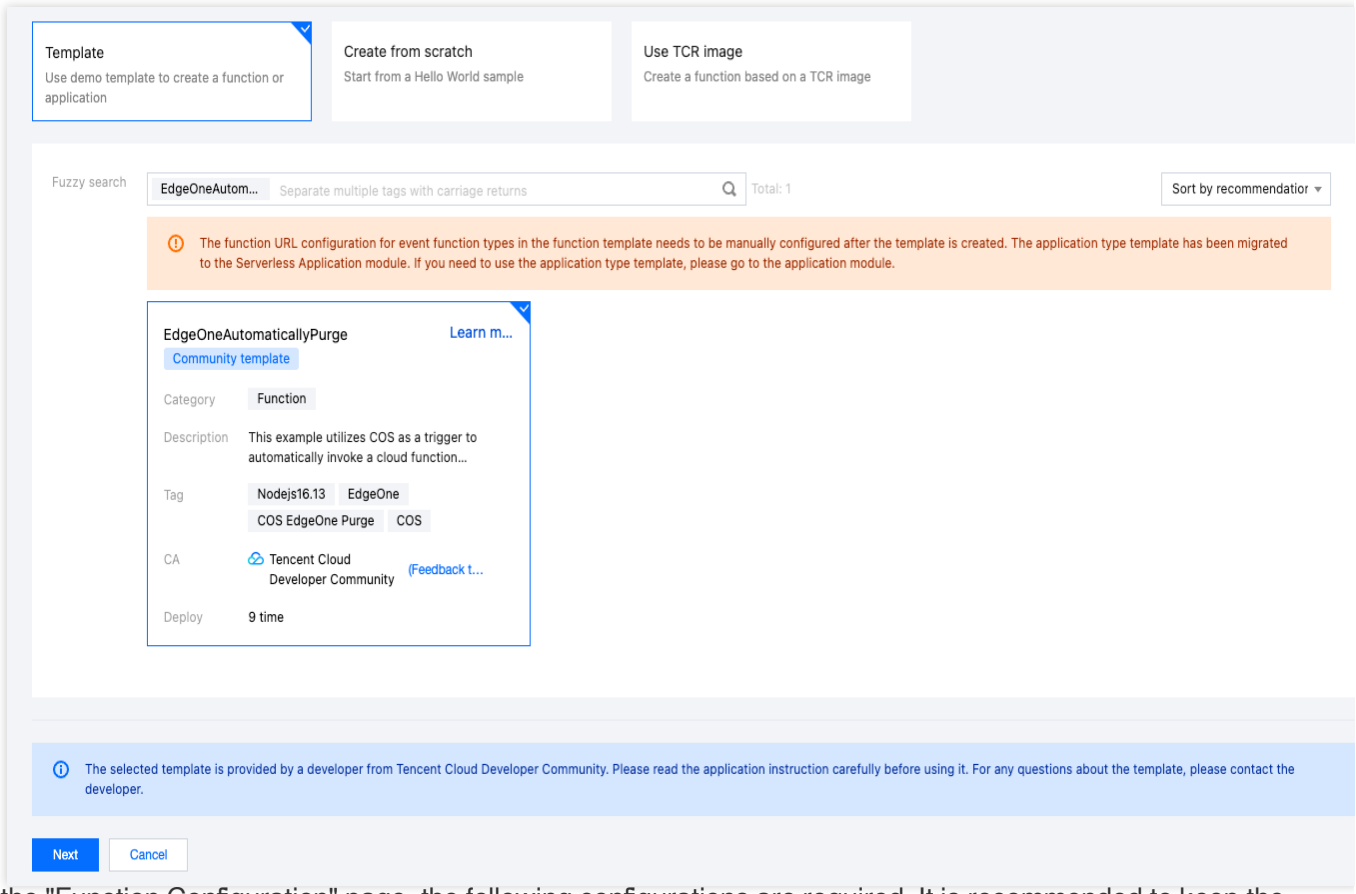
1. COS and SCF have been activated, and the bucket name and region information are recorded.
2. A site is added according to the Site Access guide, the EdgeOne package is purchased, and the site ID is obtained. The site ID can be viewed and copied in the site list after site access. For example: zone-26v607hq8d3m.



3. Addition of Acceleration Domain Name `www.example.com` has been completed at the EdgeOne console, and the origin server is configured to Tencent Cloud COS.

Step 1: Create and Deploy an SCF for Automatic Cache Purge on EdgeOne

1. Log in to the SCF Service console. In the left menu bar, click **Function Services**.
2. On the Function Services page, click **Create**, select **Template**, enter **EdgeOneAutomaticallyPurge** in the fuzzy search input box, select it, and click **Next**.



3. On the "Function Configuration" page, the following configurations are required. It is recommended to keep the default values for other configuration items.

Basic configuration

Function name: A function name will be automatically generated when the function is created. You can choose to change it to an easily recognizable function name.

Region: Select the region where the COS bucket is located. For example: Guangzhou.

Time zone: SCF uses UTC time by default. You can change it by configuring the TZ environment variable. After selecting a time zone, the corresponding TZ environment variable will be added automatically.

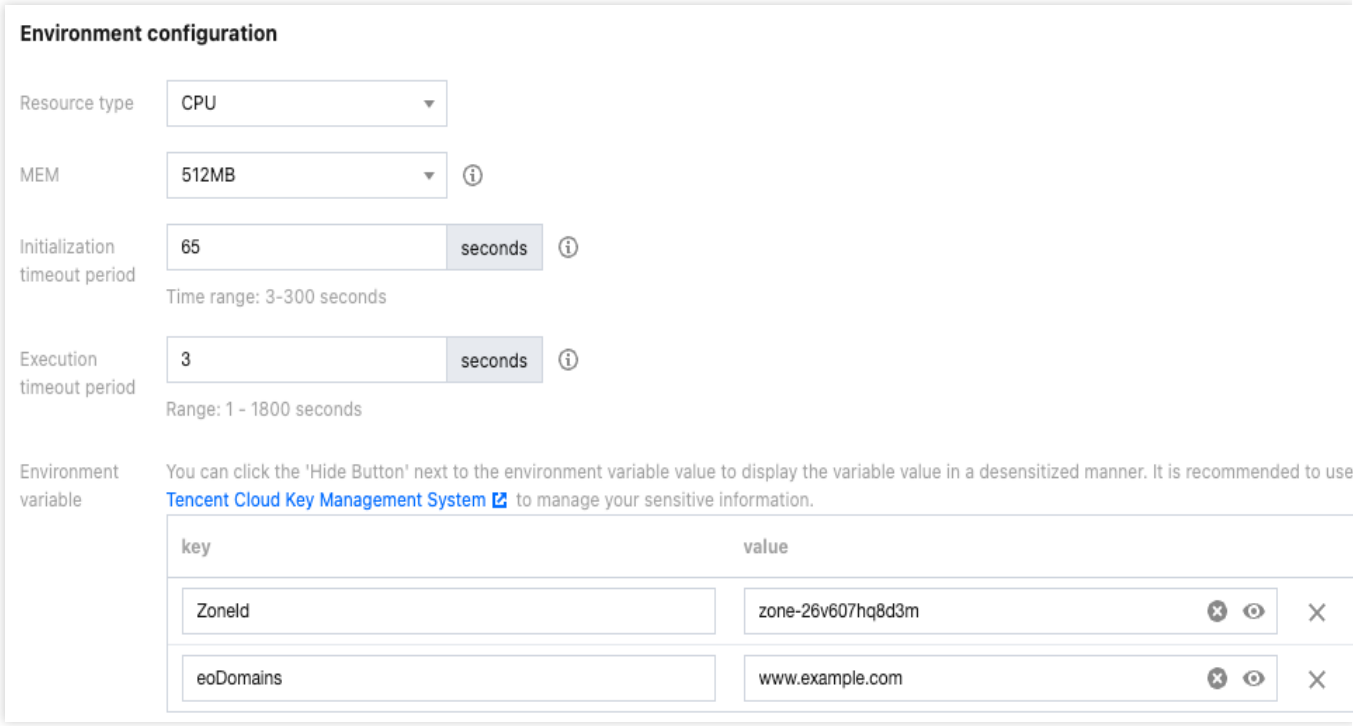
Function code: The template already includes default function code to implement EdgeOne automatic cache purge, without any modification.

Advanced configuration:

Click **Advanced Configuration**, find **Environment Configuration**, and add the following keys and corresponding values to the environment variables. Keep the rest of the configurations as default:

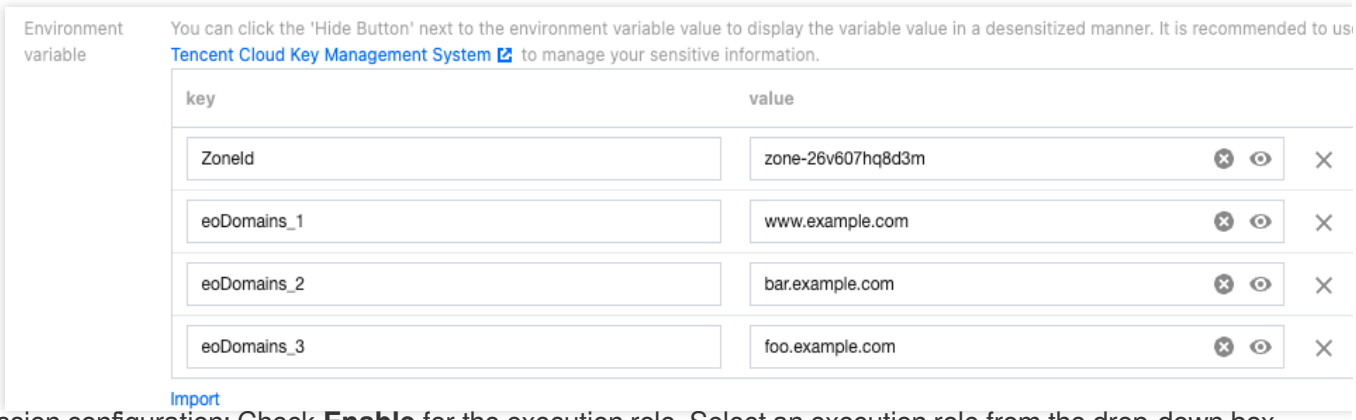
ZoneId: Enter the ZoneId of the domain name site `example.com` where automatic cache purge is required. Refer to Preparations for how to obtain the site ID.

eoDomains: Enter the acceleration domain name added under ZoneId, such as: `www.example.com` .

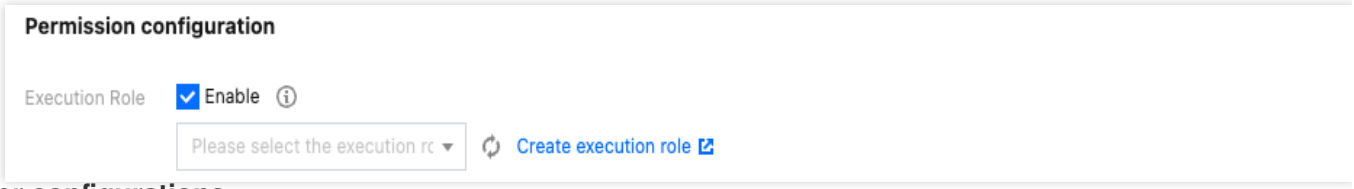


Note:

If origin servers of multiple domain names within the same site use the same COS bucket and you expect multiple domains to trigger automatic cache purge, you can add multiple environment variables starting with eoDomains when configuring the environment. For example: eoDomains_1, eoDomains_2, as shown below:

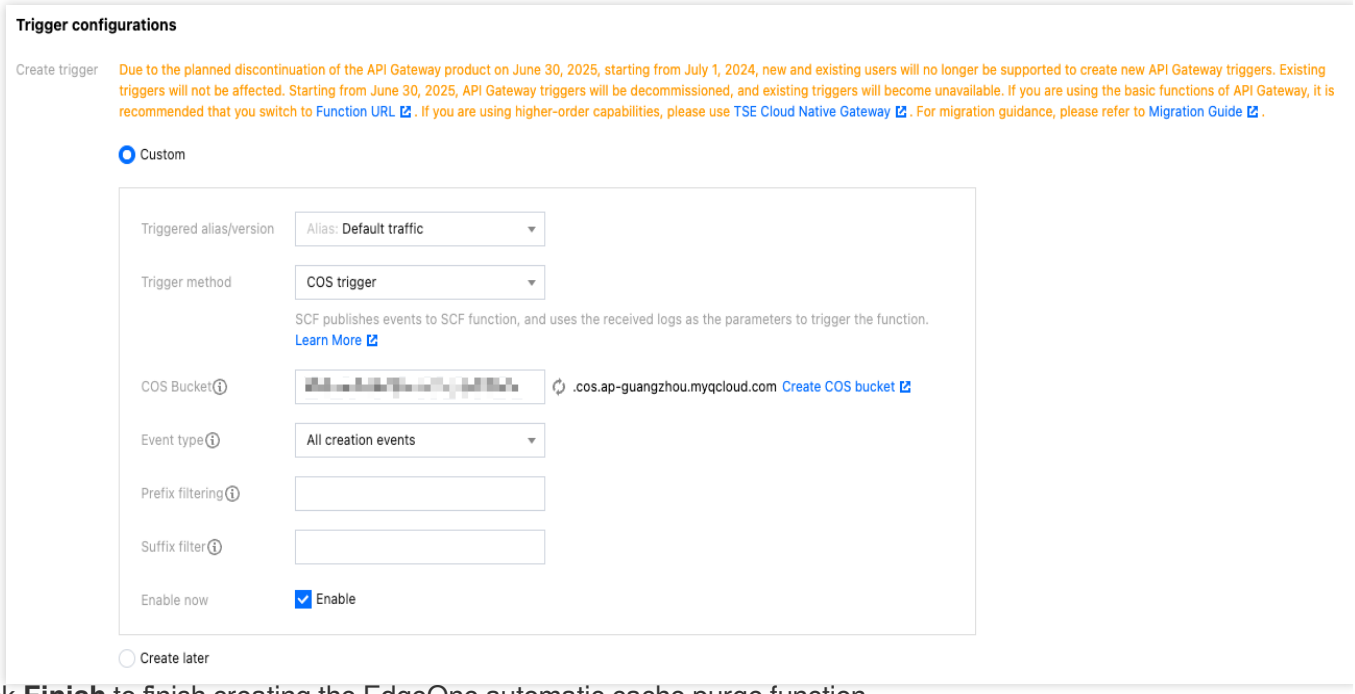


Permission configuration: Check **Enable** for the execution role. Select an execution role from the drop-down box (ensure the existing role includes the QcloudCOSFullAccess and QcloudTEOFullAccess preset policies). Otherwise, **Create execution role**.



Trigger configurations

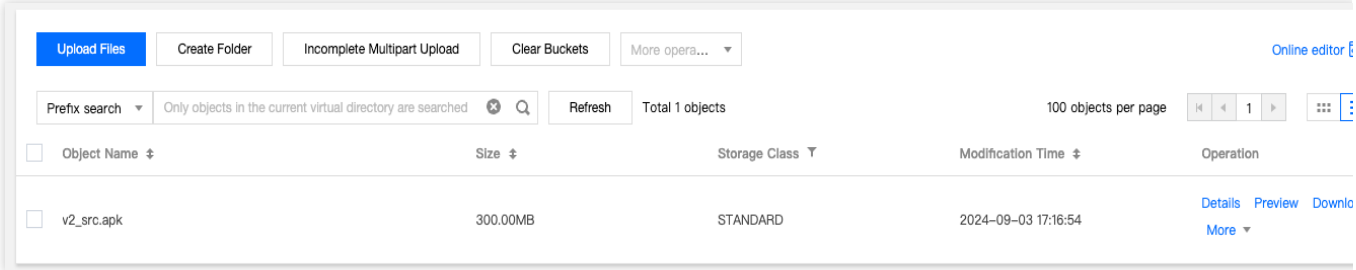
Take the `all creation events` event type as an example (if you need to create the `all deletion events` event type, refer to the configuration). In the trigger configurations, choose Custom. The COS Bucket should be in the same region as this SCF. You can input the bucket name for fuzzy search. For example: `purge-cos-1251558888.cos.ap-guangzhou.myqcloud.com` . Keep other configuration items as default.



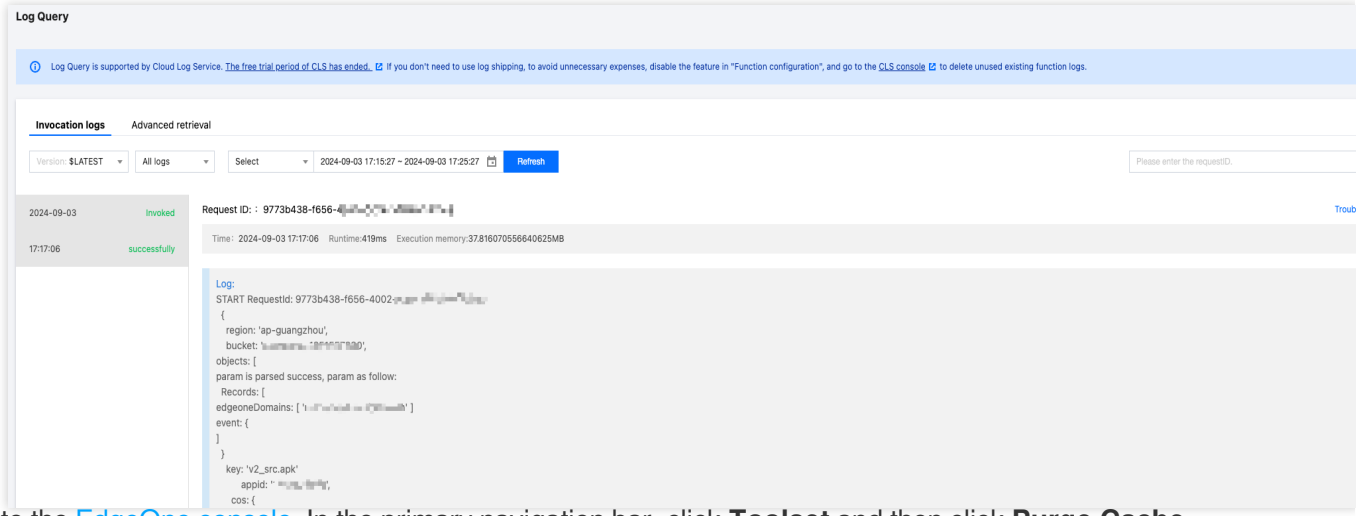
4. Click **Finish** to finish creating the EdgeOne automatic cache purge function.

Step 2: Verify the Result

1. Log in to the COS console. In the left menu, click **Bucket List**.
2. On the Bucket List page, click the **Bucket Name** used for **storing APK base packages**.
3. On the File List page, go to the root directory of `purge-cos-1251558888.cos.ap-guangzhou.myqcloud.com` .
4. Click **Upload Files**, choose a file with the same name, such as: v2_src.apk, and then click **Upload**.

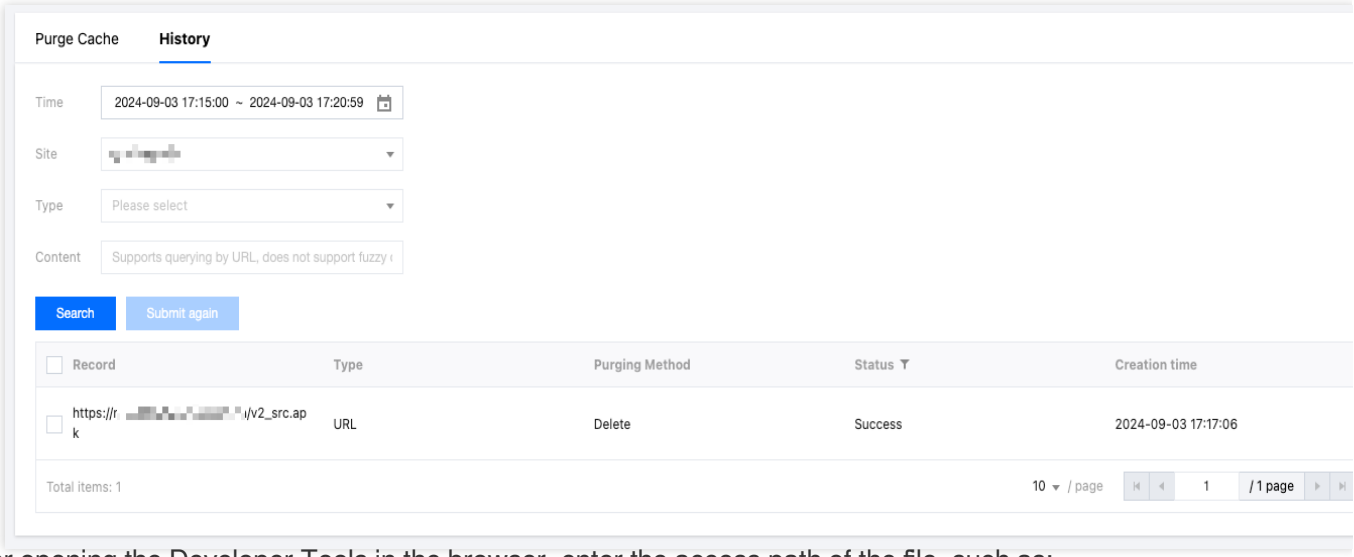


5. After the file is uploaded successfully, go to the SCF console, and click the **Function Name** in Step 1.
6. On the Function Management page, select **Log Query** > **Invocation Logs** to get the log information of the function execution. If the invocation is successful and the key information in the log matches the name of the uploaded file, it means that the file upload to COS has triggered the SCF to call the EdgeOne cache purge API successfully.

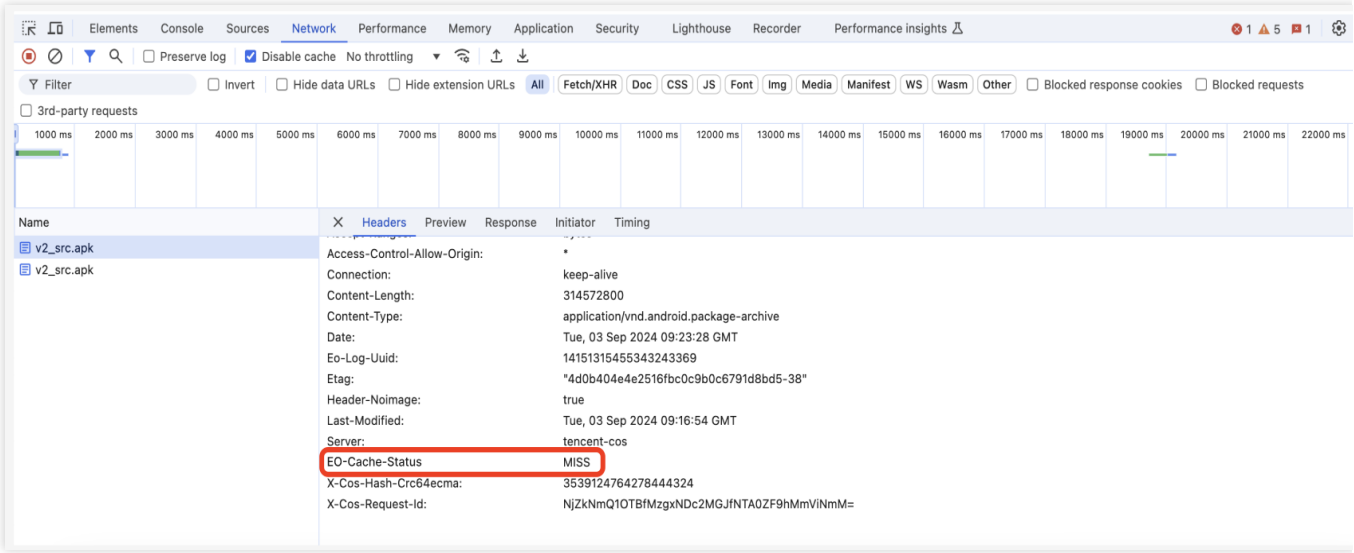


7. Go to the EdgeOne console. In the primary navigation bar, click **Toolset** and then click **Purge Cache**.

8. In the Purge Cache page, click **History** to check whether the cache purge task succeeded. If it shows that the task succeeded, it means the cache has been purged.



9. After opening the Developer Tools in the browser, enter the access path of the file, such as: `www.example.com/v2_src.apk` . Check the EO-Cache-Status value in the response header. If the cache is not purged, the old cache will be hit when the resource is accessed. As shown in the image below, if MISS is displayed, it means the resource has been cleared from the EdgeOne node, and the user request will pull the latest resource from the origin.



Monitoring Alarm (Recommendation)

EdgeOne sets quota limits for Cache Purge tasks. For specific quotas, refer to Comparison of EdgeOne Plans.
For Cache Purge tasks that exceed the quota limit, an error will be triggered when calling the Cache Purge cache API, and the final execution status of the SCF will be "calling failure". To promptly address this issue, it is recommended to configure SCF monitoring alarms through Cloud Monitor. For configuration methods, refer to: Configuring SCF Alarms, SCF Metric Descriptions.

Resource Abuse/hotlinking Protection Practical
EdgeOne Resource Abuse Prevention

Last updated：2025-01-14 11:16:59

It may take you 20 minutes to read this document, which helps you:
1. Understand what is Content Delivery Network (CDN) resource abuse and its common types and harms.
2. Understand how to configure traffic alarms and usage cap policies on the EdgeOne platform, enable real-time log push, and prevent CDN resource abuse.
3. Understand how to recognize and locate CDN resource abuse attacks by using the EdgeOne traffic analysis and log analysis features.
4. Understand the configuration recommendations in the practical tutorial for EdgeOne resource abuse prevention, which is provided for small and medium websites and enterprise-level business platforms.

What Is CDN Resource Abuse?

CDN resource abuse indicates the behavior of unauthorized users to obtain a large amount of website resources and consume the website bandwidth and server resources by illegal means. Compared to DDoS attacks, which directly affect the website availability, CDN resource abuse primarily consumes the computing resources of websites such as bandwidth and generates a sudden high bandwidth or large traffic, leading to bills higher than daily consumptions and significantly increasing the operational costs of websites. Common methods of CDN resource abuse include:
Sending a large volume of false requests through automation tools, proxy servers, or botnets;
Continuously downloading large files or transferring a large amount of data through automation tools;
Sending a large volume of concurrent requests through load testing tools to perform overload testing on the server.
Note:
If you use Tencent Cloud CDN currently, we recommend upgrading it to EdgeOne and configuring corresponding protection policies on EdgeOne. Reasonable configuration of protection measures can effectively reduce the effects of CDN resource abuse, guarantee normal business operation, and prevent unexpected high bills. For CDN service migration methods, see Guide to Using the EdgeOne Tool for Migrating Content Delivery Network (CDN) Related Services.

Preventive Measures

Configuring Usage Cap Policies

To prevent high bills caused by resource abuse attacks, it is an effective control means to add usage cap policies for key website metrics (such as bandwidth, traffic, and request volume) and configure reasonable usage caps and alarm thresholds. If an alarm occurs, you should promptly check whether the real-time requests are normal according to the Investigation Measures, and then handle the issue according to the Countermeasures.

Note:

It takes a delay of about 10 minutes for a usage cap policy to take effect. The consumption generated during this period will be billed normally.

Under all usage cap policies, the usage is calculated by subdomain name. When the effective range is selected as entire site or all subdomain names, it indicates that all subdomain names under the site will share a single cap policy.

When the same domain name has policies for two or more of the metrics such as traffic, bandwidth, and request volume, the domain name will deactivate the service if any one of these metrics reaches its threshold.

Currently, the usage cap policies can be configured only for L7 (application layer) traffic/bandwidth and HTTP/HTTPS requests, but cannot be configured for L4 (transport layer TCP/UDP application) traffic and other value-added services such as QUIC and BOT.

Configuration Sample

For detailed operations of configuring a usage cap policy, see Usage Cap Policy. In the **Add capping policy** window, select a site for the effective range and configure a cap policy based on the following suggestions:

Configuration Item	Configuration Options	Corresponding Suggestions	Use Cases
Statistical period	**5 minutes (recommended)**	Set a lower threshold to quickly detect and respond to abnormal traffic or requests.	Enables timely detection of short-term abnormal traffic or request peaks and allows you to quickly take protection measures. It is suitable for real-time monitoring and immediate response needs.
	Hour	Set a medium threshold in combination with the data from normal business peak periods to avoid false triggering of the cap during short-term traffic surges.	Captures short-term traffic fluctuation trends and provides some response time for protection adjustment.
	Day (24 hours)	Set a higher threshold based on 2-3 times the normal daily business traffic to ensure recognition of long-term abnormal traffic.	Provides a global perspective to identify abnormal traffic or request patterns throughout the day. It is suitable for formulating long-term protection policies and resource plans.
Cap	**L7 traffic**	Set a traffic threshold based on	Effectively prevents attackers

configuration	**(recommended)**	2-3 times the normal business traffic to handle traffic surges and avoid false triggering of the cap due to short-term normal traffic growth.	from consuming bandwidth resources through massive downloads of large files.
	HTTP/HTTPS request volume	Set a threshold based on 2-3 times the normal request volume to avoid false triggering of the cap during normal business peak periods.	Effectively prevents request flooding attacks that consume resources through a large volume of false requests.
	L7 bandwidth	Set a bandwidth threshold based on 2-3 times the normal bandwidth usage to handle bandwidth usage surges.	Effectively prevents excessive bandwidth consumption and avoids resource waste caused by large-traffic download attacks.
Exceeding the threshold	Disable the service, which should be enabled again in the domain list.		
Alarm threshold	**50% (recommended). An alarm message is sent when the usage reaches 50% of the configured alarm threshold.**		

Note:
When the alarm threshold is enabled and the short-term usage surge is large, since the scan interval is 5 minutes, the previous scan may not trigger the alarm threshold but the next scan directly reaches the access threshold. In this case, both a percentage alarm message and an access threshold alarm message are sent at the same time.

Enabling Real-Time Log Push

For more elaborate protection measures, it is recommended to enable the Real-time Log Push feature. This feature enables shipping access request logs to your specified destination with a low latency and supports configuration through the console or API. The latency from initiating a request to receiving the request at the destination is within 5 minutes. It is suitable for real-time monitoring and rapid investigation, such as CDN resource abuse prevention. Through real-time analysis on access behaviors, you can timely identify and analyze the characteristics of resource abuse attacks, and then configure a corresponding policy for precise blocking.
The ranges of requests recorded by different types of logs are described as follows:
Site Acceleration Logs: Record the domain name access requests, including all L7 requests passing through CDN. By default, only the allowed requests are recorded, while the blocked requests are not recorded. These logs can provide comprehensive access information and help identify abnormal high-frequency requests, abnormal traffic, and potential resource abuse behaviors.
Note:

The feature of Site Acceleration Logs in Real-Time Logs to record full L7 requests (including requests blocked by L7 protection) is currently in beta test. If you need to use it, contact us.

Rate Limiting and CC Attack Defense Logs: Record only the requests that match the security rules of the Rate Limiting and CC Attack Defense Module for L7 Protection, no matter whether the requests are blocked or not. They can help identify resource abuse behaviors through high-frequency requests.

Managed Rule Logs: Record only the requests that match the security rules of the Managed Rules Module for L7 Protection, no matter whether the requests are blocked or not. They can help detect the protection status based on managed rules and identify potential attacks and resource abuse behaviors.

Custom Rule Logs: Record only the requests that match the security rules of the Custom Rules Module for L7 Protection, no matter whether the requests are blocked or not. They can help identify abnormal requests that match custom rules and prevent specific types of resource abuse behaviors.

Bot Management Logs: Record only the requests that match the security rules of the Bot Management Module for L7 Protection, no matter whether the requests are blocked or not. They can help identify resource abuse behaviors triggered by automated scripts or malicious Bots.

Note:

Bot Management Logs are supported only after the site domain name has enabled Bot Management. For pricing details of Bot Management enabled, see VAU Fee (Pay-as-You-Go).

If you need to push specific field values in HTTP request headers, HTTP response headers, or Cookies, you can use the Custom Log Push Fields feature to accurately record such information in logs.

Investigation Measures

After configuring the preventive measures as mentioned above, if you receive an alarm and judge that the usage surge is significant, you should conduct thorough investigation at the next step. This section mainly describes how to perform multi-dimensional characteristic analysis and locating on suspected resource abuse by using the EdgeOne traffic analysis and log analysis features.

Traffic Analysis

Metric Analysis is a powerful data analysis service offered by EdgeOne, aimed to help users gain deep insights into the business operation and security status. By real-time monitoring and analysis on key metrics, you can quickly identify issues, optimize the resource allocation, and enhance the business stability and security. In scenarios of resource abuse attacks, it is recommended to focus on the following data through data filtering and selection and in combination with TOP rankings:

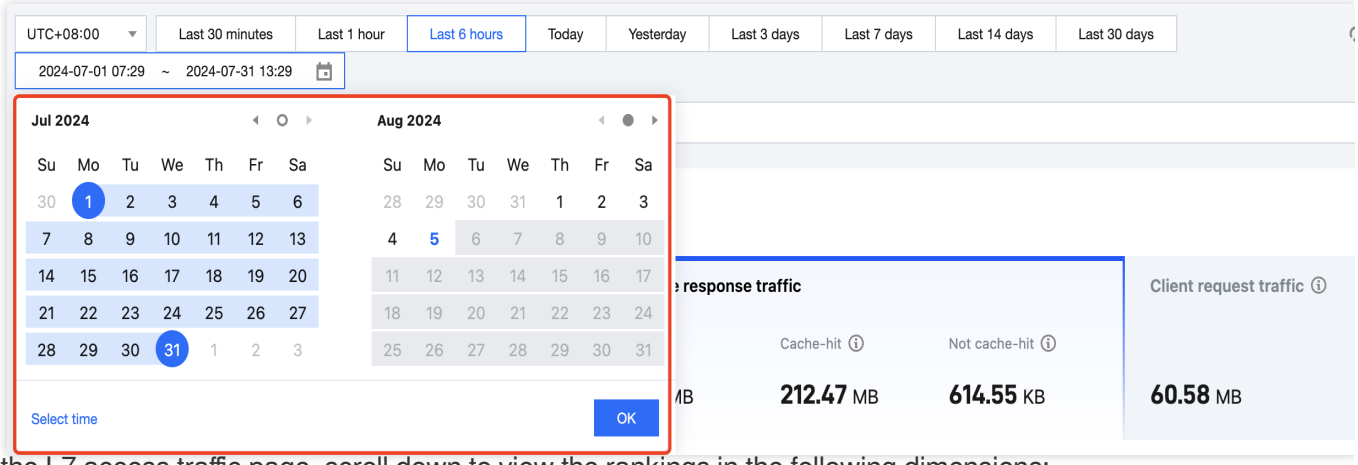
Referer distribution: High-frequency occurrence of a blank Referer or an invalid Referer often indicates credential stuffing, crawlers, or other malicious requests.

Changes in the access volumes for URLs or resource types: If the request volume for a small number of URLs or resource types surges and far exceeds that of other resources, this may indicate targeted resource abuse.

TOP client IP addresses: Observe whether most of requests come from a small number of IP addresses, and evaluate the feasibility of IP-based request frequency control.

Directions

1. Log in to the EdgeOne console and click **Metric Analysis** in the left sidebar.
2. On the metric analysis page, click **Add filter** to add sites with usage alarms into the filter.
3. Select a date range during which the suspected resource abuse attack occurs.



4. On the L7 access traffic page, scroll down to view the rankings in the following dimensions:

Hosts: Subdomain names requested by the client.

URLs: Specific URLs of resources requested by the client.

Resource types: Types of resources requested by the client, such as `.png` , `.json` , etc.

Client IP addresses: Specific source IP address of the client request.

Referers: Referer information of the client request.

Client device types:

Device type: Type of the hardware device used by the client for requests. Valid values include:

TV: Television.

Tablet: Tablet computer.

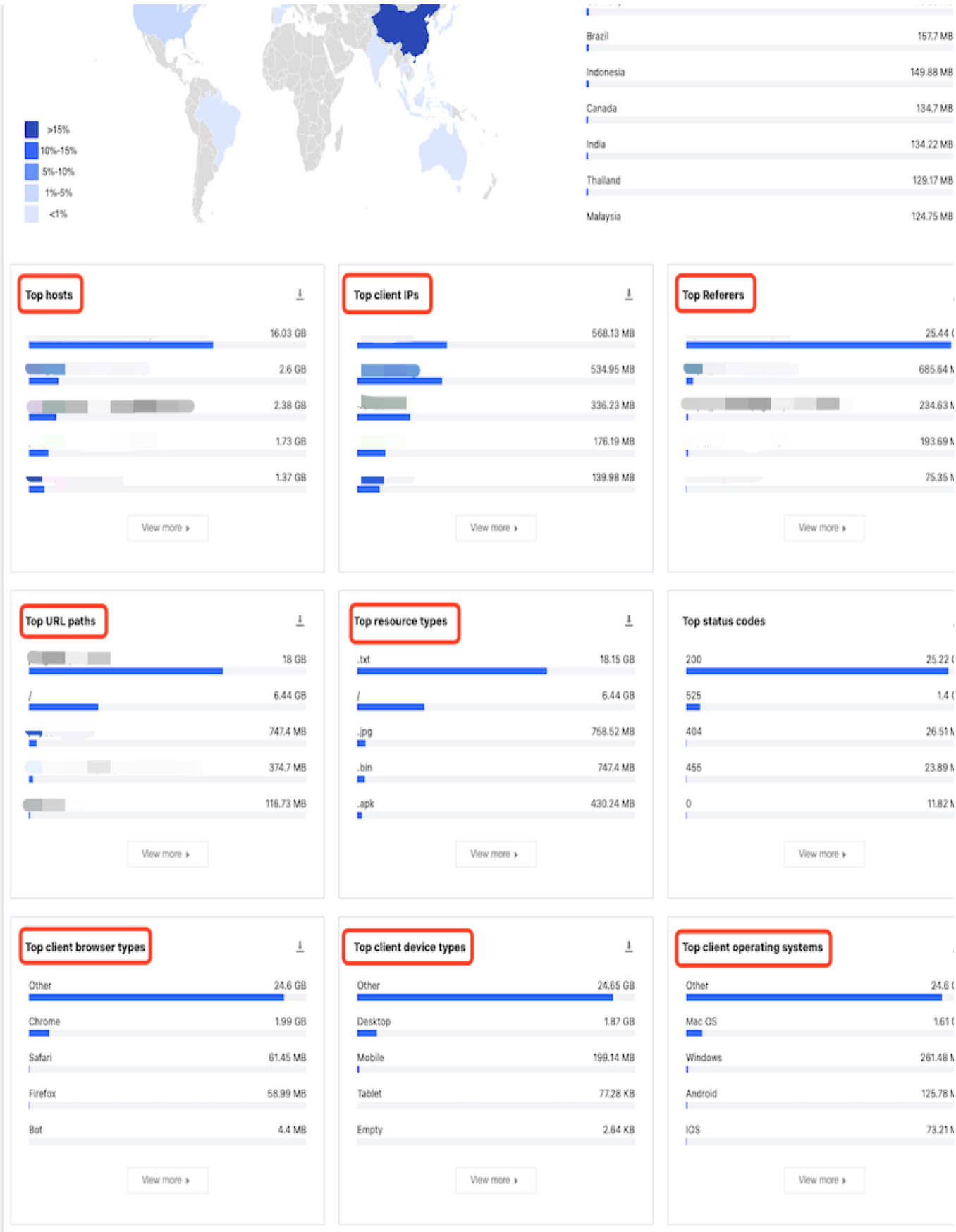
Mobile: Mobile phone.

Desktop: Computer.

Other: Others.

Browser: Type of the browser used by the client for requests.

Operating system: Type of the operating system used by the client for requests.

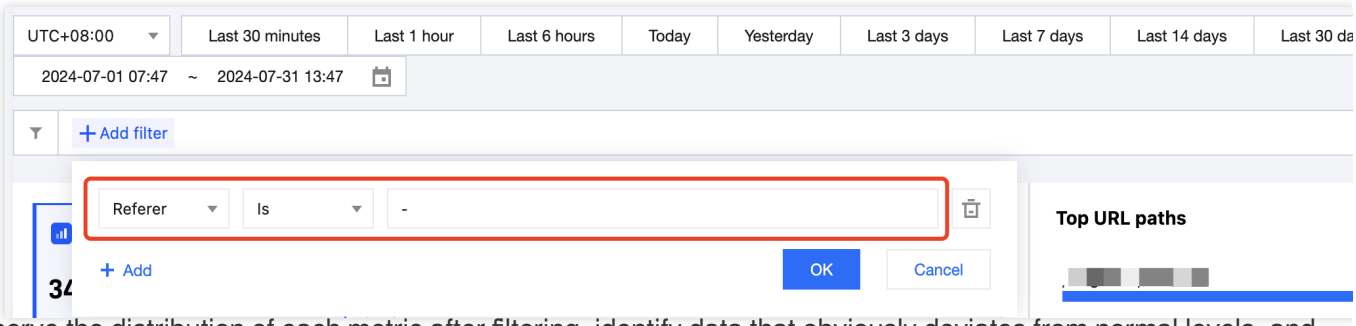


5. Click **Add filter** to add the following suggested filter criteria, which focus on abnormal traffic. Then click **OK**.

Referer: Locates requests with an empty Referer;

URL: Includes TOP 5 URLs to locate suspicious hotspot resources;

Resource types: Includes TOP 5 resource types to locate the type distribution of hotspot resources;
Client device/browser/operating system: Equals `Other; Empty` to locate suspicious abnormal clients.



6. Observe the distribution of each metric after filtering, identify data that obviously deviates from normal levels, and analyze its correlation with resource abuse.

Offline Log Analysis

To further discover more characteristics of resource abuse requests, you should perform in-depth analysis on the offline logs in the **alarm occurrence period**. Through comprehensive field analysis, we can depict the profile of resource abuse requests from multiple dimensions such as source IP, URL path, request parameter, User-Agent, and Referer source, laying the data foundation for formulating precise countermeasures at the next step. The following describes the log fields that need special attention in offline log analysis for resource abuse investigation:

Field Name	Data Type	Description	Supported by Offline Logs or Not	Supported by Real-Time Logs or Not
RequestUrl	String	URL path of the client request, excluding query parameters. This field is a key analysis dimension for resource abuse attacks.	✓	✓
RequestUrlQueryString	String	Query parameter in the URL of the client request. If the query parameter of requests for resource abuse is fixed or has obvious characteristics, you can set a blocklist for the source IP of such requests or for requests matching this parameter.	✓	✓
RequestUA	String	User-Agent information of the client request. Simple resource abuse tools often use the same User-Agent. If a large volume of access requests use a specific and uncommon User-	✓	✓

		Agent, you can consider blocking such requests.		
RequestReferer	String	Referer information of the client request. The Referer of a normal request is usually the URL of another page on the site or a search engine URL, but command line tools such as curl may forge the Referer. If the URL of the abused page is not actually referenced by other sites but appears in the Referer, it can be deemed abnormal. You can block such requests by configuring Referer Hotlink Protection.	✓	✓
ClientIP	String	Client IP connected to the EdgeOne node, namely the source IP of the request. If a small number of IP addresses far exceed other IP addresses in access volume, you can consider blocking them.	✓	✓
EdgeResponseBodyBytes	Integer	Size of the response body returned by the node to the client, in bytes. Malicious resource abuse often includes repeated downloading of large files, so analyzing the statistical results of EdgeResponseBodyBytes is a critical step in resource abuse analysis.	✓	✓

For more fields and their corresponding descriptions, see L7 Access Log Field Description.
For detailed operations of downloading offline logs, see Offline Logs.

Countermeasures

In complex and changeable attack scenarios of website resource abuse, there is no one-size-fits-all solution. EdgeOne offers a comprehensive suite of protection features including access control and rate limiting, which can be flexibly combined. You should select an optimal protection configuration combination based on the factors such as attack characteristics and actual business situations. The following provides a detailed practical tutorial for EdgeOne resource abuse prevention from different perspectives of personal site operators and online business sites.

Small and Medium Website Platforms

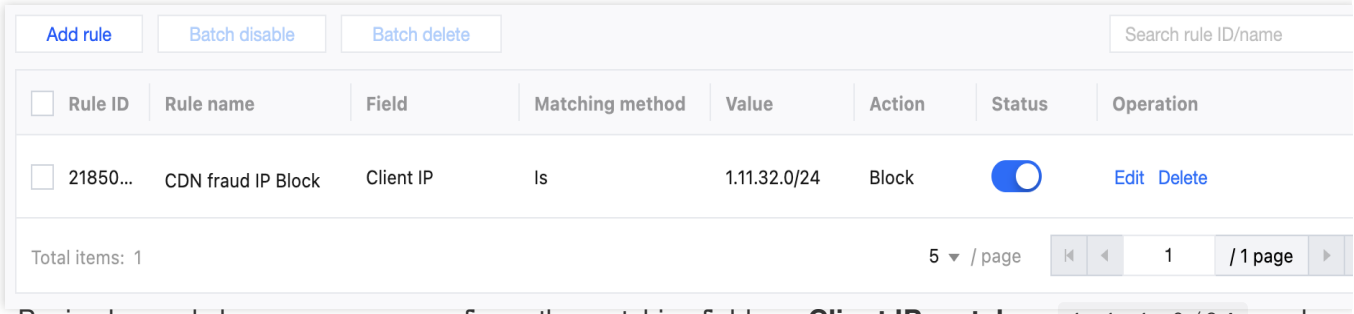
Scenario 1: Rapid Blocking of Abnormal Source IP Addresses Based on Metric Analysis

Scenario Example

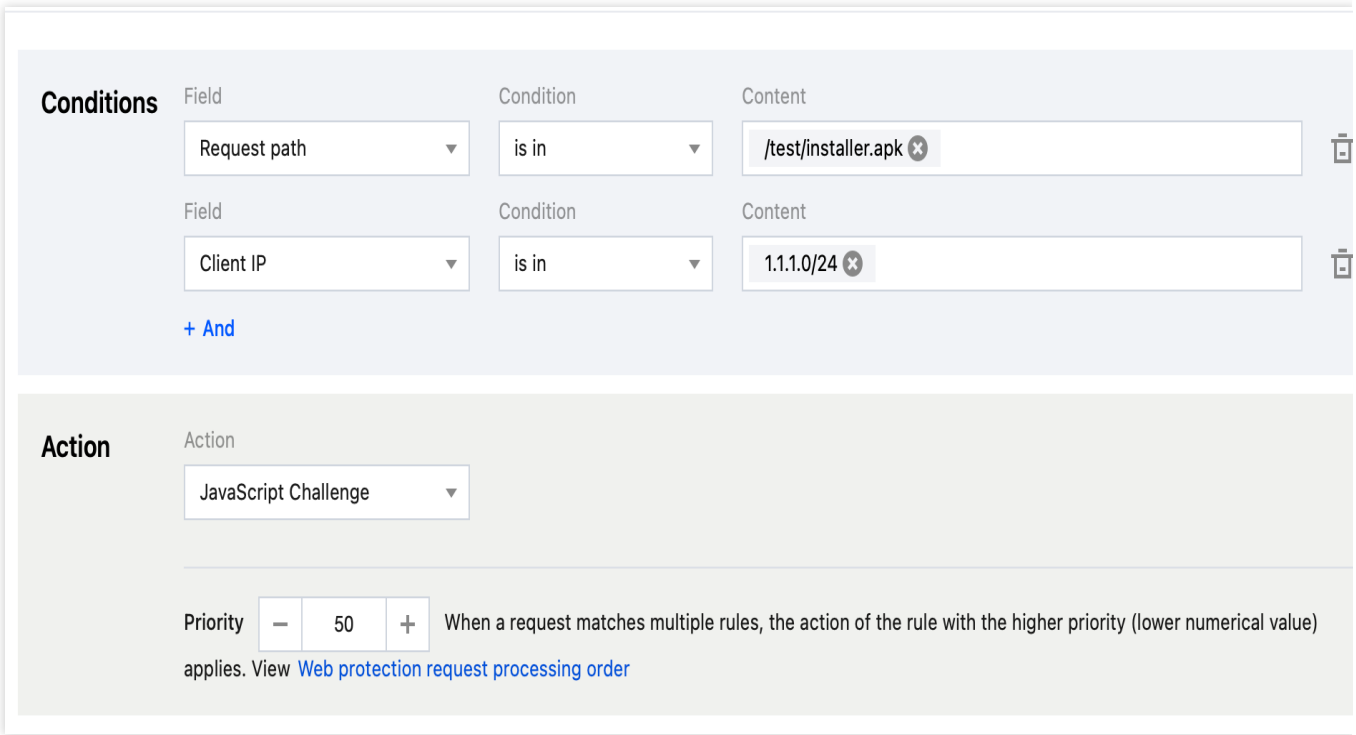
During the suspected resource abuse period, the proportion of access to a 5 MB file was found abnormally high through analysis on the resource type ranking metric of L7 access traffic. Further investigation revealed that the file path was `/test/installer.apk`, and the requests primarily came from client IP addresses of the `1.1.1.0/24` IP range. Based on this clue, you can quickly create an IP blocklist policy to block this malicious IP range and curb potential resource abuse behaviors.

Recommended Configuration

It is recommended to configure protection policies by using the custom rules of the EdgeOne Web protection feature. For detailed operations, see Custom Rules.
For the Personal plan, users can configure the rule type as **Client IP Control** in **Basic Access Control**, and select the matching method as **Client IP equals** `1.11.32.0/24` and the action as **Block**.



For the Basic plan and above, users can configure the matching fields as **Client IP matches** `1.1.1.0/24` and **Request path includes** `/test/installer.apk`, and select the action as **JavaScript Challenge** in **Precise Matching Rules**.



Scenario 2: Rapid Blocking of Abnormal User-Agents Based on Log Analysis

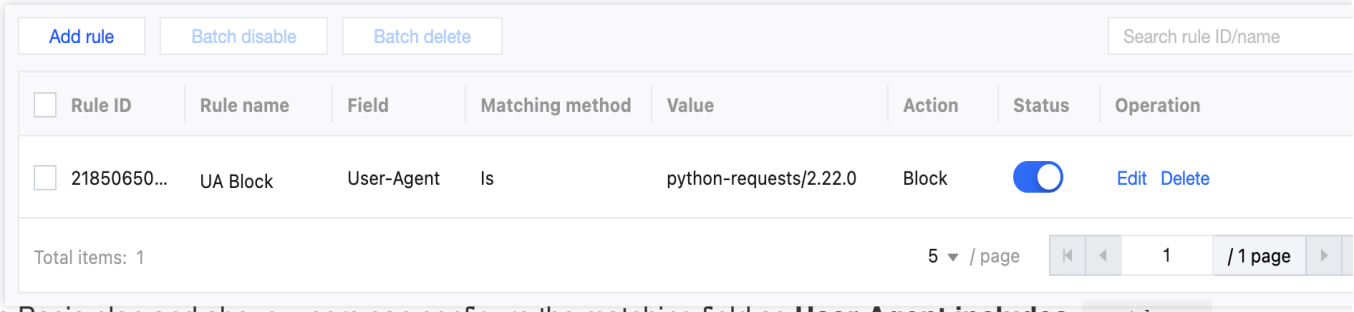
Scenario Example

Real-time logs show that within a certain period, the distribution of RequestUA was abnormally concentrated. Further analysis revealed that `python-requests/2.22.0` was accessed most frequently, and a large volume of requests used the unique User-Agent identifier of Python scripts containing `python-requests/` . Since these requests significantly deviated from the User-Agent characteristics of regular browsers, they can be identified as automated requests or even malicious crawlers. Based on this, you can configure a User-Agent blocklist rule to precisely block suspicious requests containing specific User-Agent identifiers.

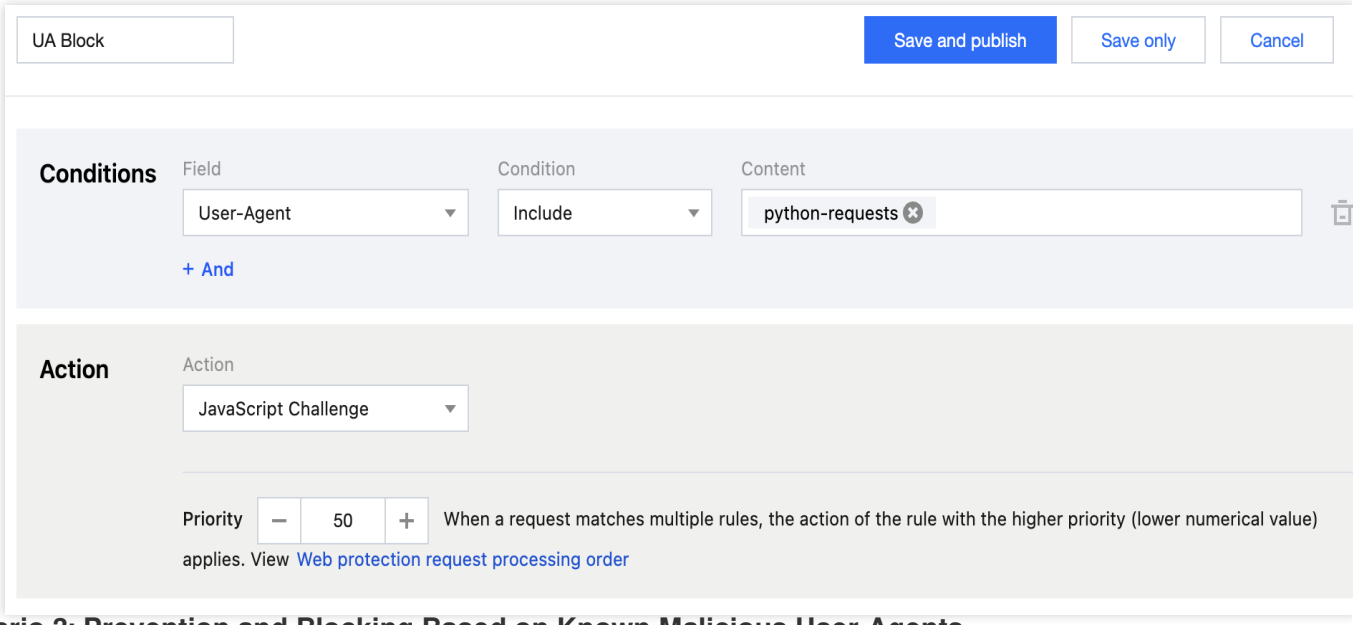
Recommended Configuration

It is recommended to configure protection policies by using the custom rules of the EdgeOne Web protection feature. For detailed operations, see Custom Rules.

For the Personal plan, users can configure the rule type as **User-Agent Control** in **Basic Access Control**, and select the matching method as **User-Agent matches wildcard pattern**, the matching content as `python-requests/2.22.0` , and the action as **Block**.



For the Basic plan and above, users can configure the matching field as **User-Agent includes** `python-requests` and the action as **JavaScript Challenge** in **Precise Matching Rules**.



Scenario 3: Prevention and Blocking Based on Known Malicious User-Agents

Scenario Example

For known common resource abuse tools, you can configure their characteristic User-Agent strings in your custom rules in advance. Preventively enabling this rule in the global site or key paths can minimize the risk of resource abuse by such tools. Common User-Agents for resource abuse include `empty User-Agent; curl/xx.xx; Wget/xx.xx; ApacheBench/xx.xx; python-requests/xx.xx` .

Recommended Configuration

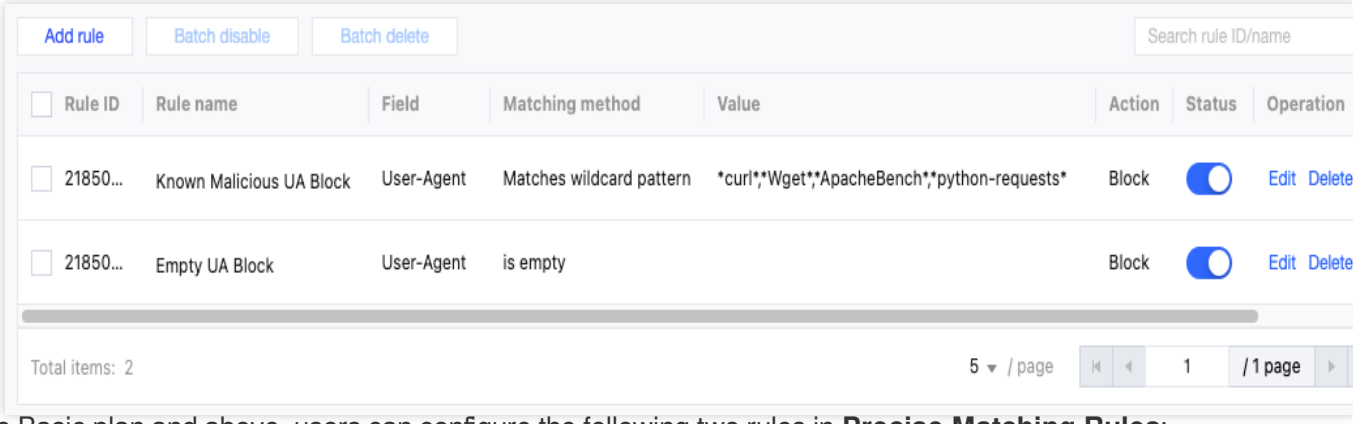
It is recommended to configure protection policies by using the custom rules of the EdgeOne Web protection feature. For detailed operations, see Custom Rules.

For the Personal plan, users can configure the following two rules in **Basic Access Control**:

Rule 1: Configure the rule type as **User-Agent control**, the matching method as **User-Agent is empty**, and the action as **Block**.

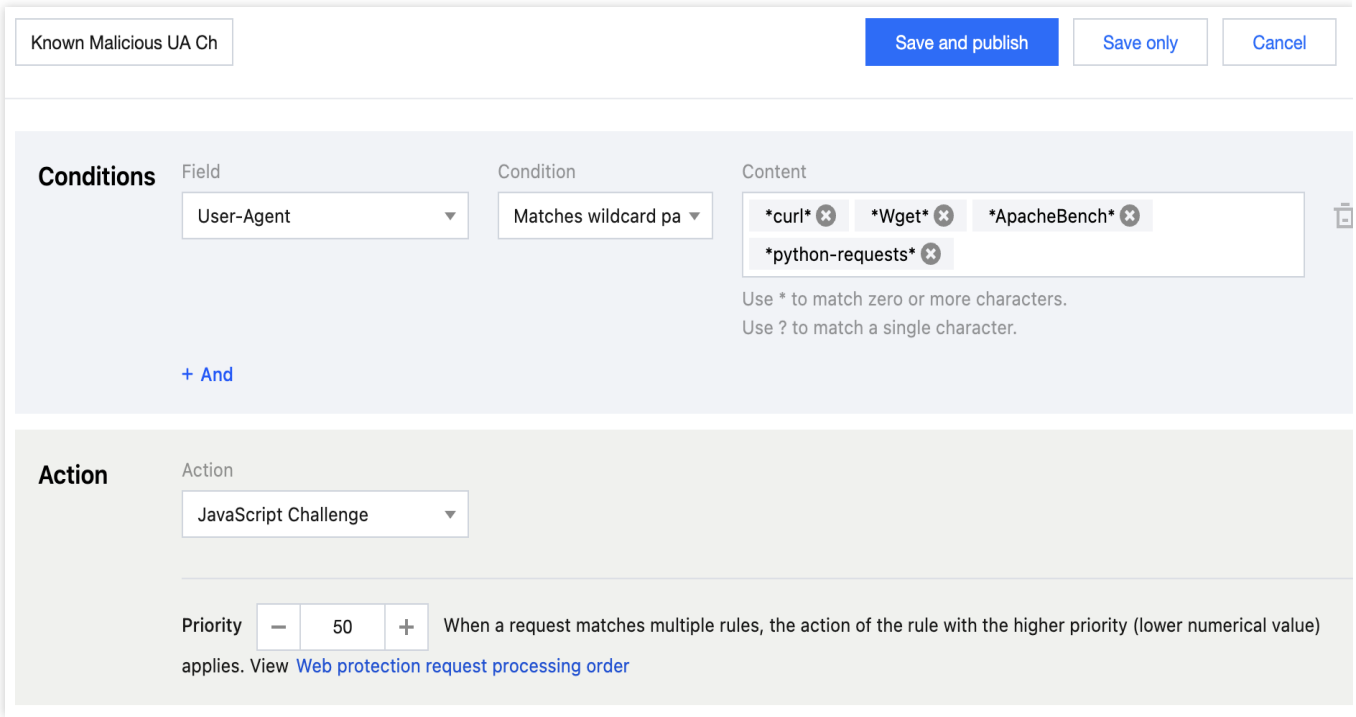
Rule 2: Configure the rule type as **User-Agent control**, the matching method as **User-Agent matches wildcard pattern**, the matching content as `curl/; Wget/; ApacheBench/; python-requests/` , and the action

as **Block**.

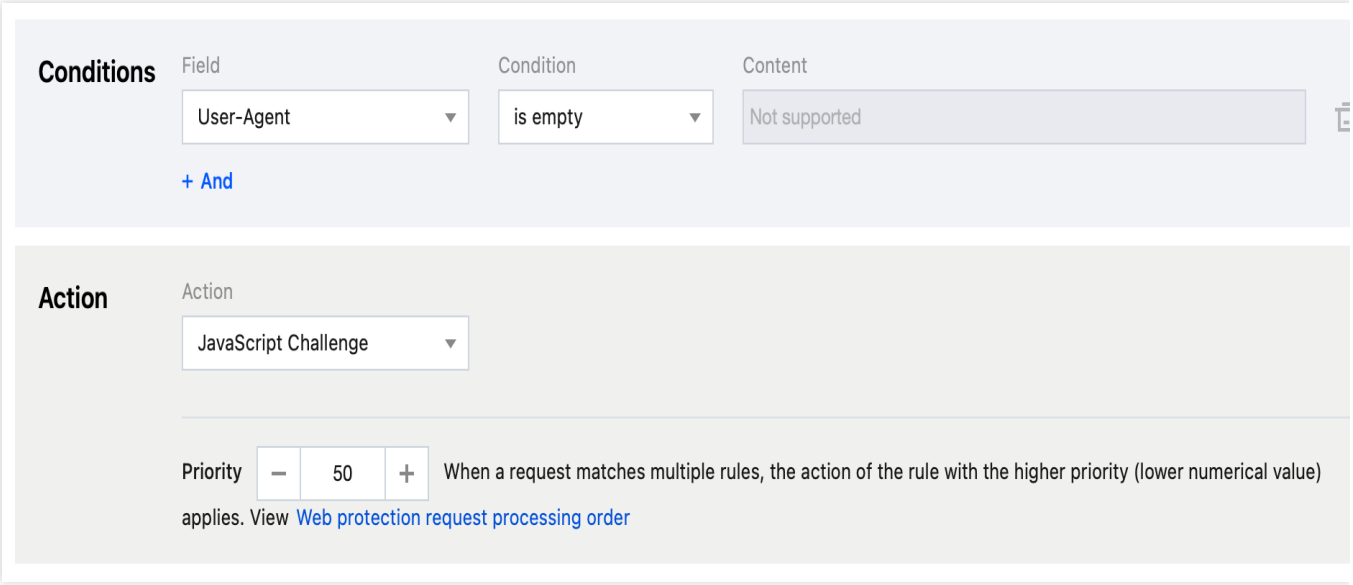


For the Basic plan and above, users can configure the following two rules in **Precise Matching Rules**:

Rule 1: Configure the matching field as **User-Agent includes** `curl/; Wget/; ApacheBench/; python-requests/` and the action as **JavaScript Challenge**.



Rule 2: Configure the matching field as **User-Agent is empty** and the action as **JavaScript Challenge**.



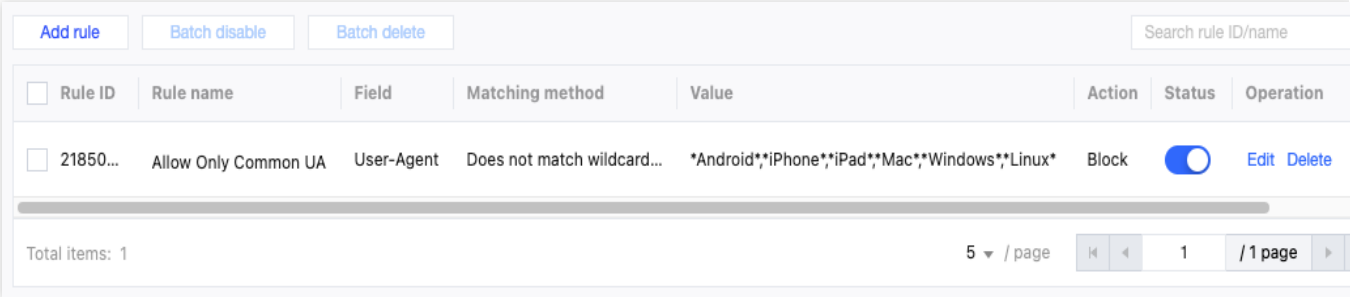
Scenario 4: Allowing Only Common User-Agents (Temporary High Defense)

In case of large-scale, dispersed User-Agent resource abuse, if it is difficult to identify each malicious User-Agent characteristic, you can use the reverse logic to allow only valid User-Agent access requests from common normal browsers and applications. This method can filter a large volume of suspicious requests at once. **However, due to the strictness of the rules, there is a risk of misjudgment, so it should be used cautiously in combination with other dimensional characteristics.**

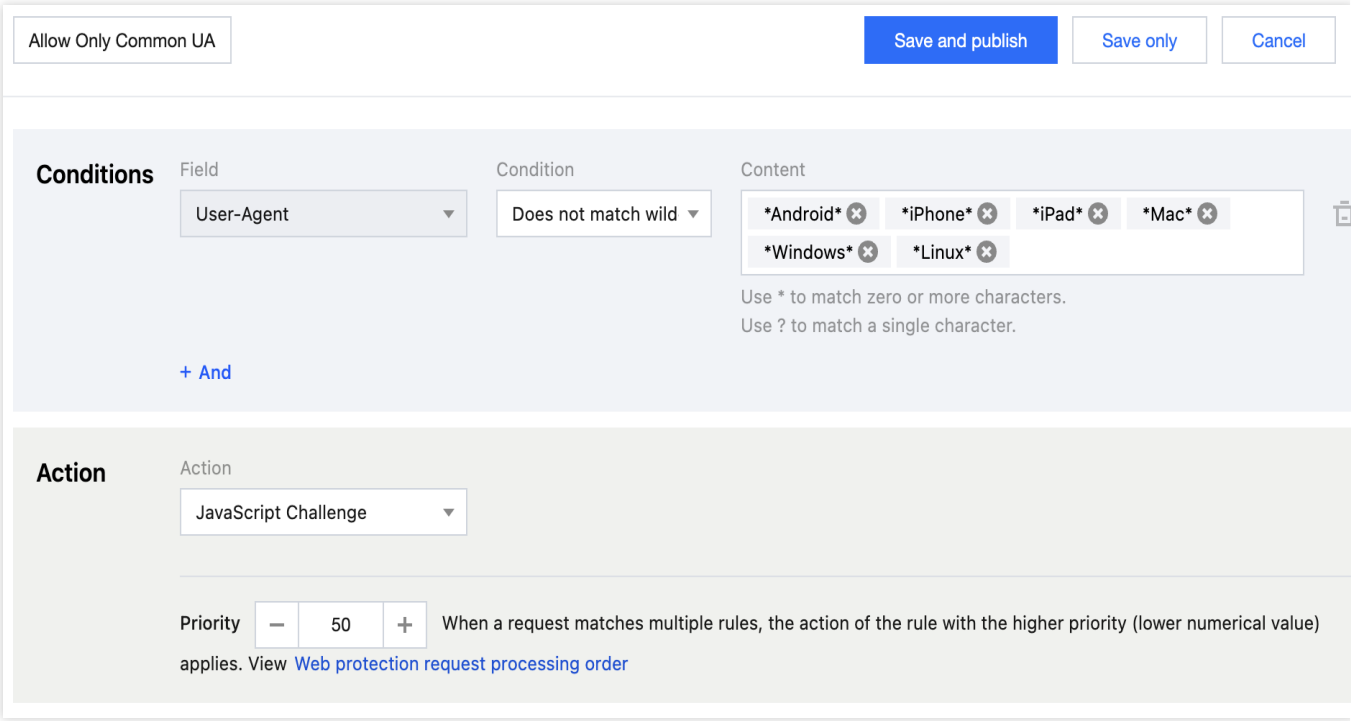
Recommended Configuration

It is recommended to configure protection policies by using the custom rules of the EdgeOne Web protection feature. For detailed operations, see Custom Rules.

For the Personal plan, users can configure the rule type as **User-Agent control** in **Basic Access Control**, and select the matching method as **User-Agent does not match wildcard pattern**, the matching content as `*Linux*; *Macintosh*; *Android*; *iPhone*; *iPad*; *Windows*` , and the action as **Block**.



For the Basic plan and above, users can configure the matching field as **User-Agent does not match wildcard pattern**, the matching content as `Android, iPhone, iPad, Mac, Windows, Linux` , and the action as `JavaScript Challenge` in **Precise Matching Rules**.



Note:

This policy is not required for application scenarios when the User-Agent is empty in normal business.

If the User-Agent value is an application name, you should add the application name of normal business in the User-Agent into the matching content.

It should be configured cautiously due to high strictness. To avoid false blocking, you should perform joint judgment in combination with other dimensional characteristics.

Scenario 5: Configuring the Single-IP High-Frequency Access Limit for CC Attacks (Temporary High Defense)

CC Attack Defense supports identifying and handling CC attacks through rate baseline learning, header feature statistical analysis, and client IP intelligence. EdgeOne offers three preset CC attack defense policies:

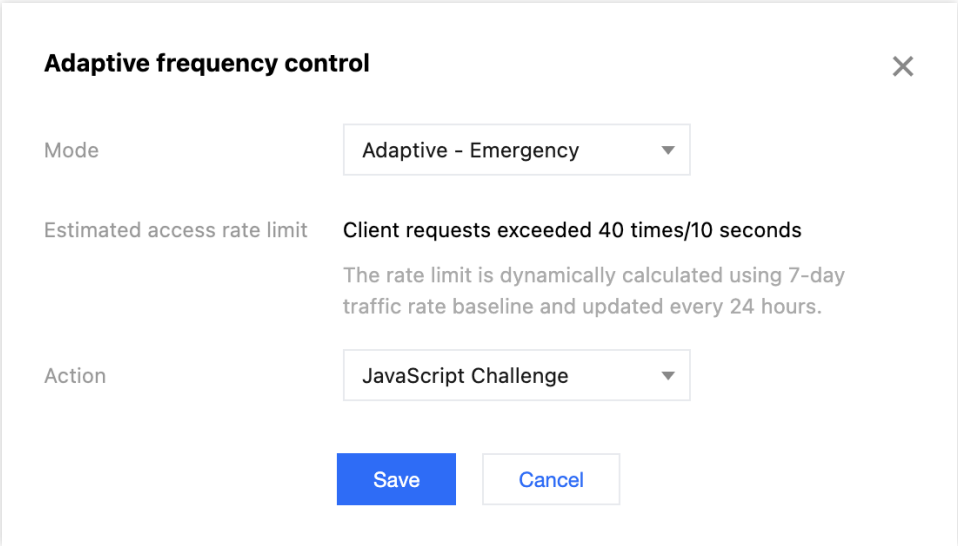
Adaptive frequency control: It is used to defend against CC attacks that occupy server resources through high-frequency and a large volume of concurrent connection requests, and can limit the access frequency based on a single IP source.

Slow attack defense: It is used to defend against CC attacks that occupy server resources through a large volume of slow connection requests, and can limit the minimum access connection rate based on a single session and eliminate slow connection clients.

Intelligent client filtering: It integrates rate baseline learning, header feature statistical analysis, and client IP intelligence, to dynamically generate protection rules in real time. It also supports human-machine verification for requests from high-risk clients or those with high-risk header features. Intelligent client filtering is enabled by default and conducts a JavaScript Challenge to clients that meet the rules.

In case of suspected website resource abuse attacks or abnormal usage alarms, it is recommended to temporarily set **Adaptive Frequency Control** to the **Adaptive - Emergency** level and the action to **JavaScript Challenge**. This

measure can efficiently block a large volume of requests from malicious IP addresses and effectively prevent resource abuse and other attacks. For detailed operations, see CC Attack Defense.



Note:

After handling the resource abuse attack, you should promptly restore **Adaptive Frequency Control** to the recommended configuration, **Adaptive - Loose**, to ensure smooth access with normal business traffic. For details of various limitation levels, see CC Attack Defense.

Scenario 6: Personalized Frequency Control Based on Business Levels

Different from strong DDoS attacks, resource abuse tends to be more covert. You should perform judgment based on specific business scenarios and formulate **personalized frequency control policies** to avoid false blocking of valid users. No matter whether the blocking policy is specific to the IP or User-Agent, it belongs to precise blocking. However, in actual attacks, the attack characteristics may not be obvious, especially when the volume of requests from a source IP address is as high as hundreds of thousands.

Defense policies combined with the business scenario first require website administrators to **evaluate the normal access mode of the business and determine the business traffic baseline**. For example, in application download or upgrade scenarios, most IP addresses are generally used for only one or two downloads. In rare cases, there might be multiple retries due to failures, but the number of retries is usually within a reasonable frequency range. Abnormal high-frequency access possibly indicates attacks or malicious resource abuse.

When a website suffers from resource abuse attacks, the domain name bandwidth significantly increases. To address this issue, you can use rate limiting of the EdgeOne Web protection feature to set thresholds based on normal business levels and configure rate limiting policies, or monitor and adjust the policies through Real-Time Logs. For detailed operations, see Rate Limiting.

Note:

During configuration, the frequency control rules should be dynamically adjusted based on the actual defense effect. Initially, you can set frequency thresholds based on empirical values to quickly achieve defense. If the effect is poor,

the rules can be gradually tightened; conversely, if the rules affect normal business, they should be appropriately loosened.

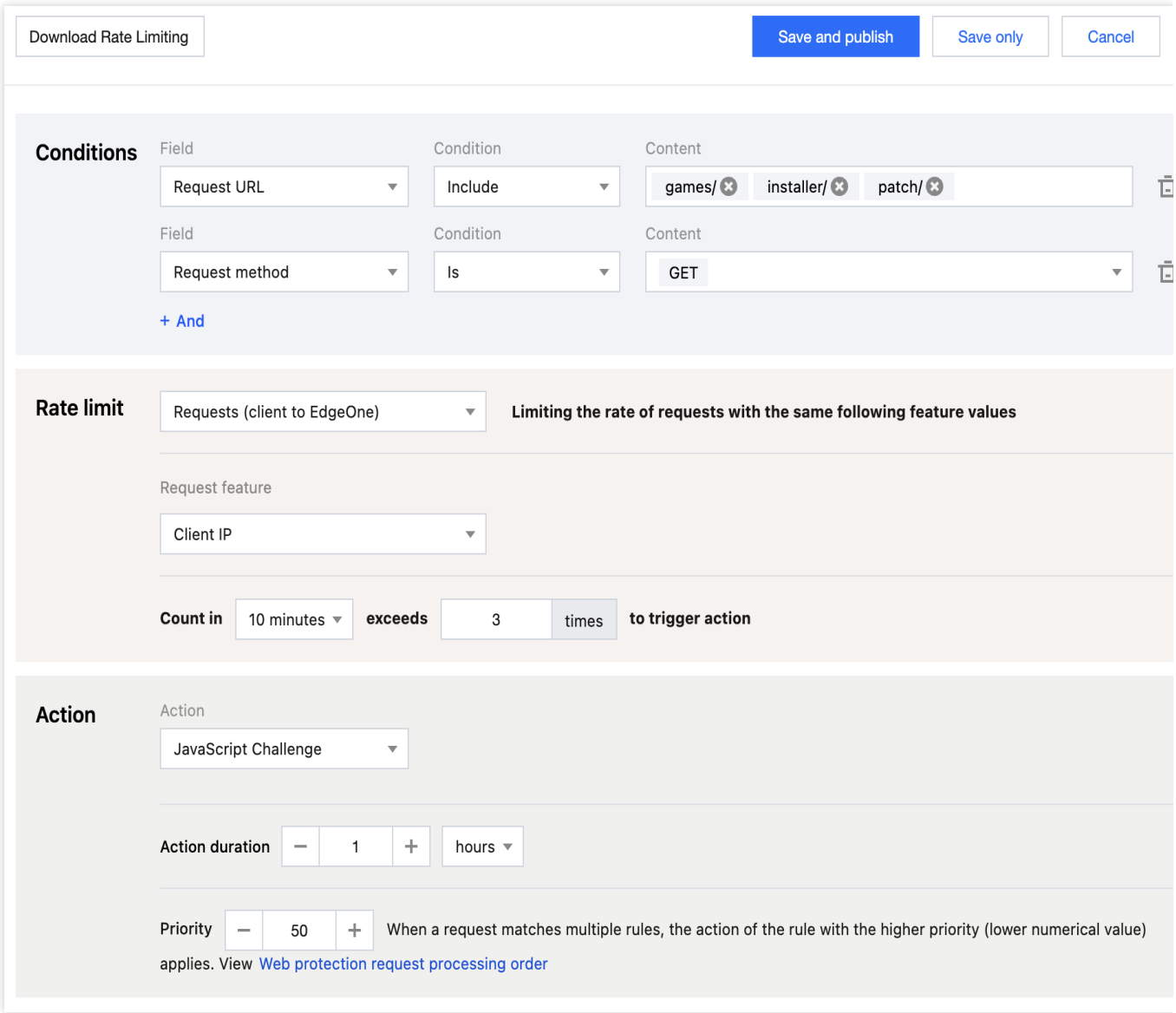
Game Package Download Frequency Limiting Based on Business Baselines

Scenario Example

A game platform offers download services for installation and update packages of multiple games through EdgeOne acceleration. The download URLs of game packages have a fixed pattern. Examples are as follows:
Game A installation package: `https://cdn.example.com/games/A/installer_v1.0.zip`
Game A update package: `https://cdn.example.com/games/A/patch_v1.1.zip`
Game B installation package: `https://cdn.example.com/games/B/installer_v2.0.exe`
Game B update package: `https://cdn.example.com/games/B/patch_v2.1.exe`
On the day when a game version was released, the number of downloads per single IP address was generally 1 and the number of download retries due to individual network issues did not exceed 3. However, the installation and update packages were downloaded frequently to some IP addresses after the version was released, far exceeding the frequency of normal user behaviors, so a usage alarm was triggered. The possible causes may be that pirate websites or sharing communities were capturing game packages, or attackers intended to consume bandwidth. You can configure the Rate Limiting rules to promptly block these malicious requests.

Recommended Configuration

In **Precise Rate Limiting**, configure the matching object as **Custom protection object**, the matching field as **Request URL includes** `games/; installer/; patch/` **and** Request method equals `GET` . In the Rate limit **section, select** Requests (client to EdgeOne) **with the request feature as** Client IP**, configure the trigger as** Count **in** 10 **minutes exceeds** 3 **times, and select the action as** JavaScript Challenge **with the action duration as** 1 hour**. For detailed operations, see Rate Limiting.



Abnormal User-Agent Frequency Limiting Based on Log Analysis

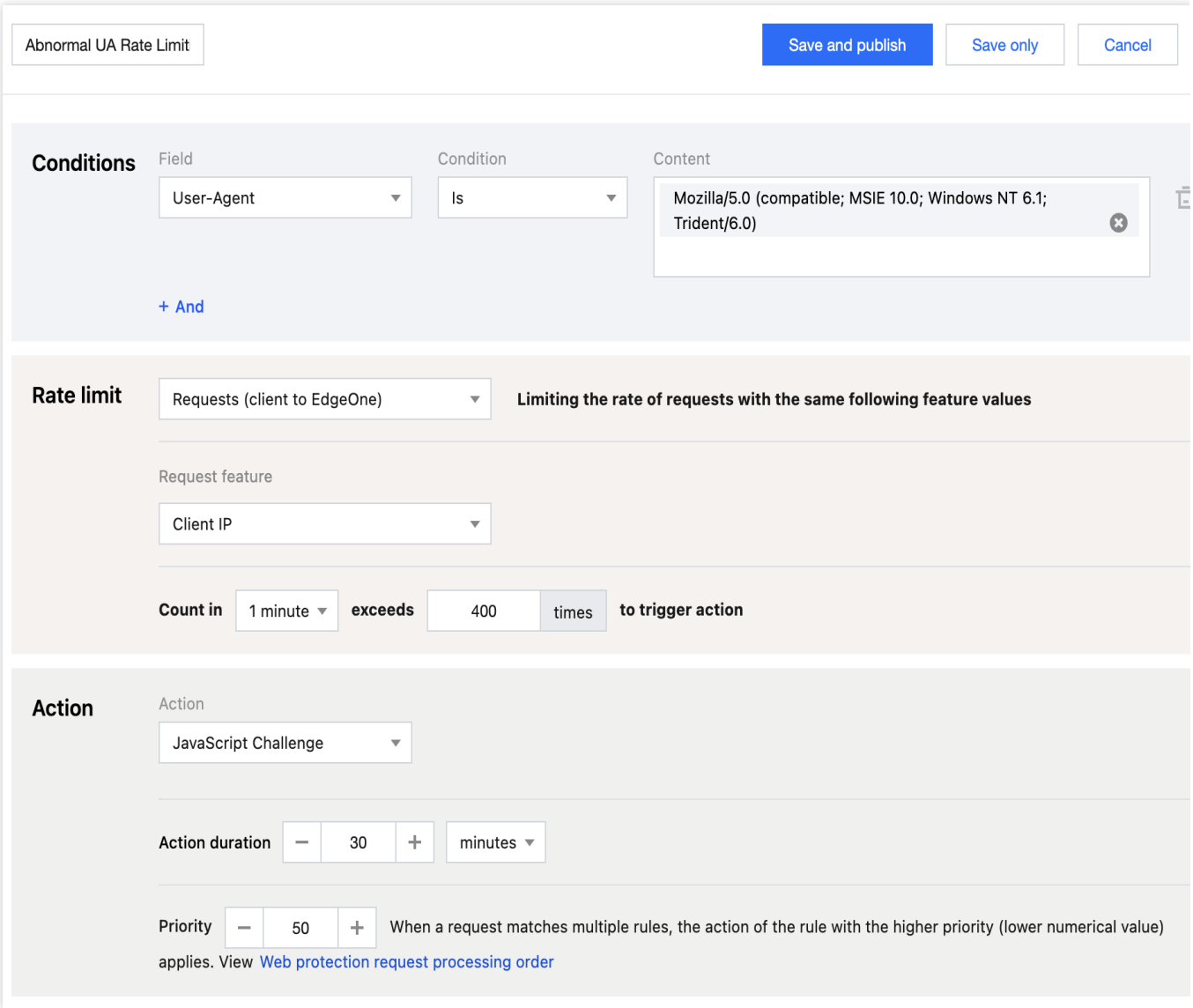
Scenario Example

A website was accessed frequently by attackers, triggering a usage alarm. Through real-time log analysis, it was found that the attacking IP addresses were dispersed, suggesting distributed attacks, but the User-Agents were abnormally uniform.

Most of requests came from a single User-Agent string, `Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; Trident/6.0)` . This case deviated significantly from normal business, which typically exhibited a diverse range of User-Agents representing various browsers and devices. Under normal circumstances, the volume of access requests with this suspicious User-Agent is very low, but at this time, the request volume surged and occupied most of the traffic. It can be basically determined as CDN resource abuse attack. You can configure the Rate Limiting rules to promptly block these malicious requests.

Recommended Configuration

In **Precise Rate Limiting Rules**, configure the matching object as **Custom protection object** and the matching field as **User-Agent equals** `Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; Trident/6.0)` . In the **Rate limit** section, select **Requests (client to EdgeOne)** with the request feature as **Client IP**, configure the trigger as **Count** in **1** minute exceeds **400** times, and select the action as **JavaScript Challenge** with the action duration as **30 minutes**. For detailed operations, see Rate Limiting.



Note:

You should adjust the threshold for triggering protection and the action duration based on your own normal business level and the attacker characteristics and attack frequency obtained from real-time log analysis.

EdgeOne Hotlink Protection Practical Tutorial

In addition to direct protection measures against resource abuse, you should pay attention to protection of the website resources and adopt active defense. Hotlink protection is an important means for preventing unauthorized use of website resources.

Hotlink refers to the behavior of illegally referencing and using resources (such as images, videos, software packages, etc.) from the original site on other sites and consuming the bandwidth and resources of the original site, without the permission of the website owner. It not only infringes the legal rights of the original site but may also cause adverse SEO impacts on the original site. Therefore, it is necessary to actively implement hotlink protection measures.
EdgeOne offers a perfect hotlink protection solution to control hotlink behaviors by Referer hotlink protection, Token hotlink protection, remote authentication, etc. This protects your content from unauthorized hotlink access and enhances the security of acceleration services. For details, see EdgeOne Hotlink Protection Practical Tutorial.

Enterprise-Level Business Platform

For online business sites facing resource abuse threats, in addition to using general protection measures commonly adopted for personal sites, it is recommended to select the EdgeOne Standard or Enterprise plan and enable the Bot management feature. With its built-in artificial intelligence engine and extensive behavioral feature analysis, you will gain a smarter, more effortless Bot management experience, to effectively address various resource abuse attacks.
The Bot Intelligence Analysis module employs advanced machine learning algorithms to form a threat identification model through massive data training. This model extracts key behavioral characteristics from multiple dimensions such as request rate, IP intelligence, URL sequences, and SSL/TLS fingerprints, and adopts techniques such as cluster analysis and similarity comparisons to accurately determine whether a request comes from an automated program and has malicious intents. It can reduce false blocking of valid requests with a comprehensive and multi-dimensional analytical approach.
Additionally, the EdgeOne Enterprise plan supports JA3 fingerprint characteristics. Website administrators can preset fingerprint conditions for high-risk Bots based on their business scenarios, to achieve precise blocking of specific attack tools. For example, you can incorporate fingerprints of Python libraries and headless browsers commonly used by malicious crawlers into the resource abuse defense rules, to automatically block related traffic and achieve more proactive and efficient protection.
Note:
The Bot management feature is supported only after the site domain name has enabled Bot Management. For pricing details of Bot Management enabled, see VAU Fee (Pay-as-You-Go).

EdgeOne Hotlink Protection Practical Tutorial

Last updated：2025-03-21 11:51:28

This document describes how to use the hotlink protection capabilities provided by EdgeOne to protect your content against unauthorized hotlinking and improve the security of acceleration services.

Background

Hotlinking refers to other websites or applications directly linking to your resources without your authorization. This behavior can seriously impact your website. First, hotlinking consumes your bandwidth and server resources, causing your website to slow down and even potentially crash the server. Second, hotlinking may lead to your content being misused or spread without authorization, which can seriously damage your brand image and reputation.
To address these issues, EdgeOne provides a range of powerful Hotlink protection capabilities. By using EdgeOne's Hotlink protection feature, you can ensure that only authorized users can access and use your content. You can set an allowlist to permit only specific domain names or IP addresses to access your resources, thereby preventing unauthorized hotlinking. Additionally, you can flexibly configure custom Hotlink protection rules as needed.
With EdgeOne's Hotlink protection capabilities, you can effectively safeguard your content. You can confidently provide high-quality content without worrying about hotlinking and misuse issues. This helps maintain your brand image and reputation while saving bandwidth and server resources, enhancing your website's performance and reliability.

Implementation Method

HTTP response: Implements basic access control such as IP blocklist/allowlist, Referer blocklist/allowlist, UserAgent blocklist/allowlist, and regional access control. For details, see HTTP Response.The issues with this method are:
IP addresses can be forged or hidden, allowing attackers to bypass IP allowlist and blocklist restrictions. They can use proxy servers, Virtual Private Networks (VPN), or other technologies to hide the actual IP address, thereby bypassing access restrictions. This makes IP allowlist and blocklist less reliable in preventing unauthorized access;
The Referer header can also be easily forged. Attackers can bypass Referer allowlist and blocklist restrictions by modifying the Referer field in the HTTP request header. They can use browser plugins, proxy tools, or other technologies to alter the Referer field, making it appear as if it comes from a trusted source, thereby bypassing access restrictions;
The issue with the User-Agent header is similar to the Referer header, as it is also easily forged.
Token authentication: Timestamp hotlink protection, which is more secure and reliable. For details, see Token Authentication.Compared to the basic access control mentioned above, its advantages and issues are as follows:

Prevent link reuse: Each link contains a timestamp parameter. Even if the link is forwarded or shared, once the timestamp expires, others cannot use the link to access the resource.
Difficult to forge: Timestamp Hotlink protection increases the difficulty of hotlinking because attackers need to know the authentication algorithm, authentication key, timestamp format, etc., to construct a URL that can pass verification. These details are hard for attackers to guess or forge.
Client modification required: The use of this feature requires cooperation between the client and EdgeOne. After the client initiates an encrypted URL request, EdgeOne is responsible for Legitimacy Verification of the URL based on predefined rules. Therefore, some additional overhead and complexity need to be considered during implementation.
Edge functions: Customizable hotlink protection capabilities such as remote authentication can be supported through Edge Functions.The advantages and issues of this method are as follows:
High security: Remote authentication can provide higher security. The authentication process is completed by a remote server provided by the customer, rather than the EdgeOne node server, reducing the risk of being compromised by attackers.
Flexibility and scalability: Remote authentication provides greater flexibility and scalability. Customers can flexibly control the authentication logic to adapt to changing business requirements and user access patterns.
Customer modification: Customers need to deploy a remote authentication service and ensure the reliability, security, and performance of the remote server to avoid increased authentication delays, which could affect service quality. Additionally, to handle possible exceptions, an appropriate authentication timeout should be set. Generally, if authentication times out, the request is directly allowed. However, if the authentication service encounters an exception, it may result in illegal requests being allowed, thereby increasing security risk.

Directions

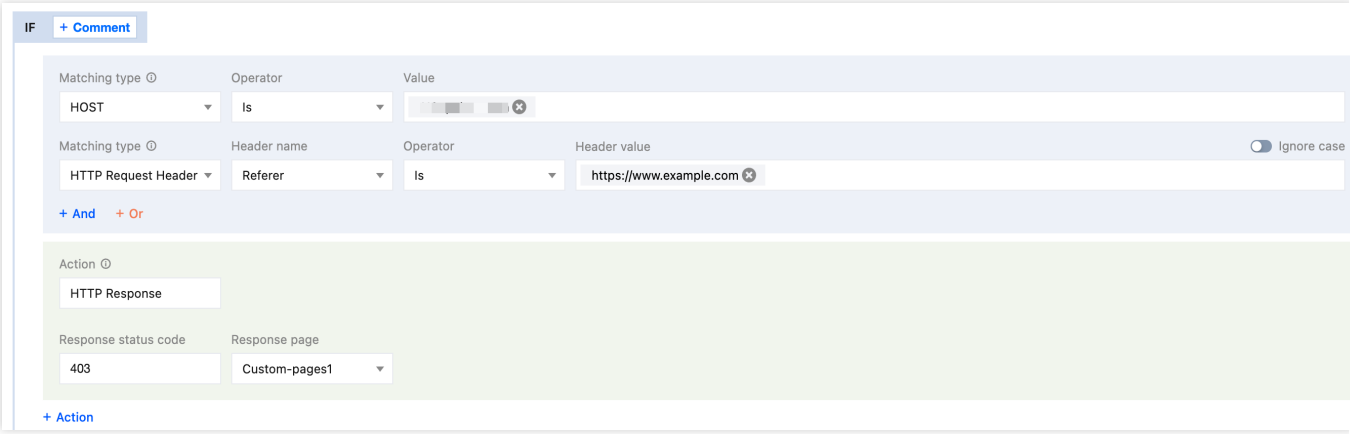
Referer Hotlink Protection

Setting access control rules based on the Referer field in the HTTP request header helps identify and filter visitors, preventing illegal use of website resources. After the Referer blocklist/allowlist is configured, EdgeOne will authenticate requests based on the list, allowing or denying access requests. If the request is allowed, EdgeOne will return the resource link; if denied, EdgeOne will return a 403 response code.

Configuration Samples

For your site `example.com`, if you only allow access to the domain business `www.example.com` with the Referer set to `https://www.example.com`, and deny other requests directly with a 403 response, you can follow these steps:
1. Log in to the EdgeOne console and click **Site List** in the left sidebar. Subsequently, in the Site List, click the **Site** you want to configure.
2. On the site details page, click **Site Acceleration** to enter the global configuration page for the site. Then, click the **Rule Engine** tab.

3. On the Rule Engine page, click **Create rule**, and then select **Add blank rule**.
3.1 On the rule editing page, set the matching type to HOST equal to `www.example.com`, and set the matching type HTTP Request Header Referer's header value not equal to `https://www.example.com`.
3.2 Click **Action**, and in the pop-up operation list, select the operation as **HTTP Response**.
3.3 Configure the response status code as 403. Select the response page from the drop-down list. If no page is available, you need to click **Create Page** to create one first and then reference it.
4. The complete rule configuration is as demonstrated below. By clicking **Save and publish**, the rule configuration will be completed.



IP Blocklist/Allowlist

By configuring the IP blocklist/allowlist to filter user requests, you can intercept or allow access from specific IP addresses, effectively limiting access sources and addressing issues such as hotlinking by malicious IP addresses and attacks.

Configuration Samples

For your site `example.com`, if you only allow access from client IP addresses within the range of 1.1.2.1 to 1.1.2.254 (including 1.1.2.1 and 1.1.2.254) to the domain business `www.example.com`, and deny other access directly with a 403 response, you can follow these steps:
1. Log in to the EdgeOne console and click **Site List** in the left sidebar. Subsequently, in the Site List, click the **Site** you want to configure.
2. On the site details page, click **Site Acceleration** to enter the global configuration page for the site. Then, click the **Rule Engine** tab.
3. On the Rule Engine page, click **Create rule**, and then select **Add blank rule**.
3.1 On the rule editing page, set the matching type to HOST equal to `www.example.com`, and set the matching type client IP equal to `1.1.2.0/24`.
3.2 Click **Action**, and in the pop-up operation list, select the operation as **HTTP Response**.
3.3 Configure the response status code as 403. Select the response page from the drop-down list. If no page is available, you need to click **Create Page** to create one first and then reference it.

4. The complete rule configuration is as demonstrated below. By clicking **Save and publish**, the rule configuration will be completed.

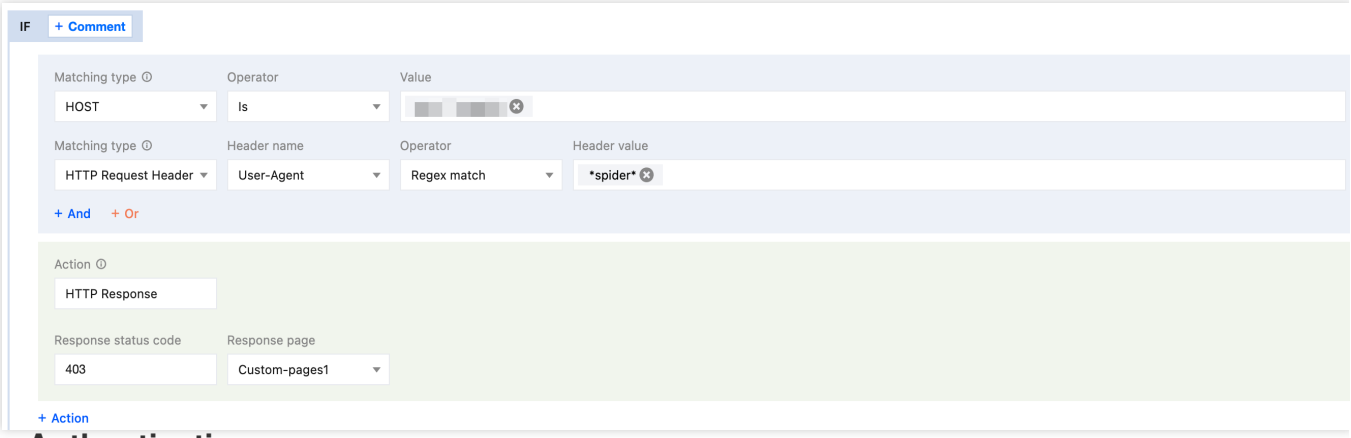
UserAgent Blocklist/Allowlist

User-Agent is part of the HTTP request header, which identifies the operating system and version and the browser type and version used by the user for accessing. You can configure the User-Agent blocklist/allowlist to restrict the sources of users accessing business resources and enhance the security of acceleration.

Configuration Samples

The domain business `www.example.com` under your `example.com` site is maliciously crawled by Google crawlers, causing a sudden bandwidth increase and severely impacting fees. Through analysis, it was found that the crawler request's User-Agent contains `spider` . If you want to block such requests, you can follow these steps:

1. Log in to the EdgeOne console and click **Site List** in the left sidebar. Subsequently, in the Site List, click the **Site** you want to configure.
2. On the site details page, click **Site Acceleration** to enter the global configuration page for the site. Then, click the **Rule Engine** tab.
3. On the Rule Engine page, click **Create rule**, and then select **Add blank rule**.
3.1 On the rule editing page, set the matching type to HOST equal to `www.example.com` .
3.2 Click **Action**, and in the pop-up operation list, select the operation as **HTTP Response**, and set the matching type to HTTP Request Header User-Agent with a regular expression match for the header value `*spider*` .
3.3 Configure the response status code as 403. Select the response page from the drop-down list. If no page is available, you need to click **Create Page** to create one first and then reference it.
4. The complete rule configuration is as demonstrated below. By clicking **Save and publish**, the rule configuration will be completed.



Token Authentication

Token authentication is a simple and highly reliable access control policy. By configuring authentication rules for URL access validation, it can effectively prevent malicious hotlinking of site resources. The use of this feature requires cooperation between the client and EdgeOne. The client is responsible for initiating encrypted URL requests, and EdgeOne is responsible for legitimacy verification of the URL based on predefined rules. For detailed configuration and usage, refer to Token Authentication.

Remote Authentication

If you have your own authentication server, you can configure remote authentication to forward user requests to the authentication server you specify. The server then validates the requests. This method is suitable for scenarios requiring precise access control and real-time authentication. EdgeOne can achieve remote authentication capability through edge functions. For sample functions, refer to Remote Authentication.

HTTPS Related Practices
Quickly Enabling HTTPS Access with EdgeOne Free Certificate

Last updated：2025-07-02 10:49:15

This document introduces how to use the free certificate service provided by EdgeOne, to help your website quickly achieve HTTPS access and reduce the workload of subsequent certificate updates and maintenance.

Background

HTTPS access has become a mainstream demand on the Internet. It can ensure secure data transmission for you to access websites, preventing such problems as information leakage and message hijacking. Additionally, in search engines, websites not enabling HTTPS are identified by the browser as unsecure websites and their search weights are also affected. Therefore, it is essential for websites to enable HTTPS access.

To achieve HTTPS access, you should find an appropriate free certificate authority (CA) to apply for a free certificate or purchase a more credible paid certificate. The following challenges exist:
1. **Complex application process:** Certificate application needs to be completed separately for each domain name and requires DNS validation or HTTP validation based on the CA's requirements. If there are a large number of domain names, DNS entries must be added for each domain name one by one to complete validation. The workload is relatively high.
2. **High deployment and maintenance costs:** After the certificate application is completed, you should deploy the certificate yourself on the server. If there are many certificates, you should deploy and maintain a correct certificate for each domain name to avoid HTTPS access errors. The update and maintenance workload is high.
3. **Prone to expiration:** Certificates must be renewed before expiration, otherwise HTTPS access alarms will occur. Especially for free certificates, the current validity period is 3 months generally, so frequent renewals are required.
4. **High costs of paid certificates:** Although the number of paid certificates can be reduced by applying for a wildcard domain name certificate and auto-renewal is supported, paid certificates are unsuitable for small websites or businesses with many domain names due to high costs.

Solution Strengths

The free certificate service provided by EdgeOne simplifies the implementation of HTTPS access, eliminating the cumbersome process of manual application, deployment, and maintenance of certificates. You can enable HTTPS for

your websites with simple operations and enjoy auto-renewal and additional access acceleration and security protection services. Compared to purchasing paid SSL certificates or applying for free certificates from other authorities, it has the following advantages:

1. **Simple application:** You only need to click **Apply for Free Certificate** in the console, and EdgeOne will automatically complete the subsequent certificate application and validation process.
2. **Easy deployment:** Once the certificate application is completed, the certificate will be automatically issued and deployed, without the need to manually download and deploy the certificate.
3. **Auto-renewal:** The free certificate can be automatically renewed, without the need for manual maintenance, so as to avoid failure of HTTPS access to websites due to certificate expiration.
4. **Additional services:** After accessing EdgeOne, your site not only enables HTTPS access, but also obtains access acceleration and security protection capabilities, further enhancing the website access experience.

Certificate Type	EdgeOne Free Certificate	Paid SSL Certificate	Free Certificate Applied For by Yourself
Fees	**Free**	Requires additional payment.	Free
Application method	**Automatic application and validation**	Requires DNS validation or HTTP validation during the application.	Requires DNS validation or HTTP validation during the application.
Deployment mode	**Automatic deployment**	Supports quick deployment within the same cloud resource and requires manual deployment for other resources.	Requires manual deployment.
Update method	**Automatic update**	SSL certificates purchased from Tencent Cloud can be automatically renewed/updated after hosting. Certificates from other sources require manual updates.	Method 1: Apply for a free certificate to manually update it before expiration. Method 2: Maintain a code script to achieve automatic application/update of free certificates.
Issuance speed	**Issued immediately after validation.**	1 business day or above, depending on the certificate type.	Issued immediately after validation.
Certificate credibility	General	High	General

Note:

1. Free Certificates are issued by the TrustAsia and Let's Encrypt. If your site is currently accessed through NS, you can apply for a wildcard domain name certificate. If it is accessed through CNAME, EdgeOne only supports the

application of single domain name certificates and does not support the application of wildcard domain name certificates.
2. The certificate has a validity period of 90 days. The platform will automatically apply for renewal 15 days before expiry, so there is no need for you to manually update it. If you are currently using NS access and switch to CNAME access, the applied wildcard domain name certificate will not be able to auto-renew upon expiration.
3. Free certificates do not support downloading.
4. If the domain is accessed via CNAME or DNSPod hosted access, you need to complete the CNAME configuration and wait for the CNAME status to take effect before applying for a free SSL certificate for the domain. In the CNAME or DNSPod hosted access mode, EO will apply for the free certificate through HTTP verification. During the verification process, the EO node will directly respond with the verification value. It is recommended to avoid using line/region-based resolution when configuring CNAME records, as this may lead to difficulties in obtaining the correct verification value, resulting in the failure of the free certificate application.

Sample Scenario

For example, the current website plans to use the services of 5 domain names including `example.com`, `www.example.com`, `api.example.com`, `image.example.com`, and `video.example.com`, all of which require enabling HTTPS access. Below is a comparison of the differences in the HTTPS access implementation paths between accessing and not accessing EdgeOne.

Not Accessing EdgeOne

When EdgeOne is not accessed, for implementing HTTPS access to websites, you should register a domain name, deploy the origin server services, and then choose a suitable CA to apply for the specified certificate. If there are multiple domain names, you should apply for a separate certificate for each domain name, or directly purchase a wildcard domain name certificate, and then deploy the certificate and enable the HTTPS service on each origin server separately, so as to achieve HTTPS access.

Before the certificate expires, you should renew it by applying for a new certificate from the CA in advance, and then update and redeploy it on the server. If there are a large number of domain names, HTTPS access errors may occur due to untimely certificate updates. Therefore, more maintenance work is required for HTTPS certificates.

Accessing EdgeOne for Free Certificates

After domain name access to EdgeOne, you can apply for a free certificate through EdgeOne, to automatically complete certificate application, issuance, and deployment and quickly achieve HTTPS access. Your origin server does not need to deploy an HTTPS certificate, and HTTP access can still be used for origin-pull.

Before the certificate expires, EdgeOne will automatically renew the certificate and deploy it to EdgeOne, saving you a lot of maintenance work.

Directions

1. Refer to Quick Start to complete site access and domain name access.
2. After domain name access, if your site uses CNAME access, you should complete CNAME configuration for your domain name and wait for the CNAME status to take effect; if your site uses NS access, you should complete modifying DNS servers and wait for the resolution to take effect. Then proceed to the next step.
3. In Domain Name Management, select `example.com` and click **Edit** in the HTTPS Configuration column. In the pop-up window for HTTPS certificate configuration, select **Free certificate** and then click **OK**.

4. After the application is completed, issue and deploy the free certificate.
5. After deployment is completed, visit the current site again to achieve HTTPS access.
6. Repeat steps 2-4 for the domain names `www.example.com`, `api.example.com`, `image.example.com`, and `video.example.com` to apply for a free certificate in a similar manner.

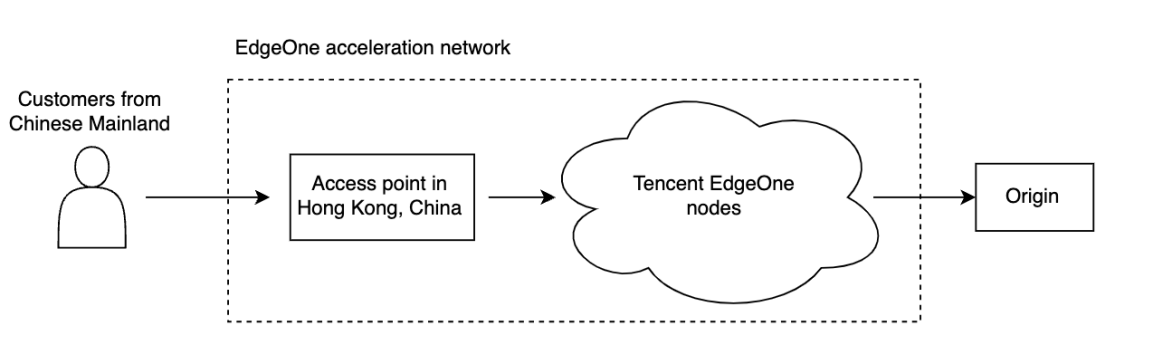
Acceleration Optimization
Cross-regional Secure Acceleration (Oversea Sites)

Last updated：2024-10-11 16:04:09

The Cross-MLC-border acceleration function leverages EdgeOne's global nodes, offering cross-regional secure acceleration solutions for service providers.

Background Introduction

A certain Web service is deployed overseas and provides services to the public through `www.example.us` (overseas site). It is temporarily unable to be hosted on servers within the Chinese Mainland due to its overseas location. This poses challenges for the service as its main customer base is located in the Chinese Mainland, resulting in network issues such as delays, jitter, packet loss, and the risk of interruptions. To optimize the user experience for Chinese Mainland users, EdgeOne provides the Cross-MLC-border acceleration function, which leverages the Hong Kong access point and Tencent Cloud acceleration network to effectively solve the problems faced by cross-regional services.



Prerequisites

1. Follow the site access guide to add a site, purchase the EdgeOne Enterprise plan, and set the site acceleration area to Global (MLC excluded).
2. Contact the business department to enable the Cross-MLC-border acceleration function.

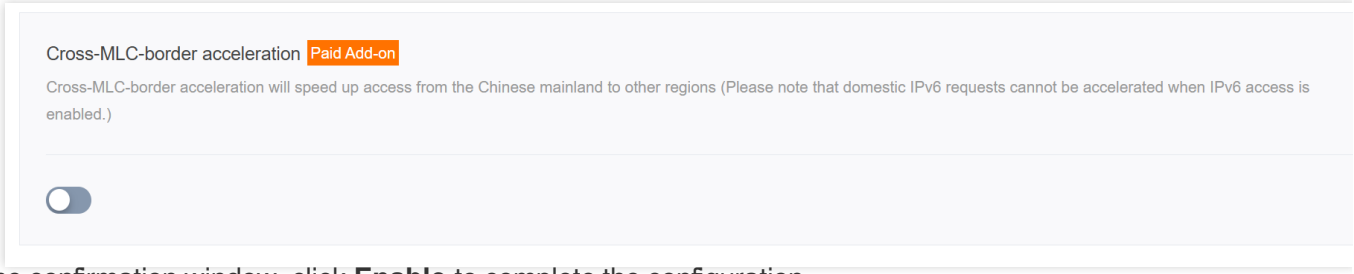
Note：

1. This function is only supported by the EdgeOne Enterprise plan.

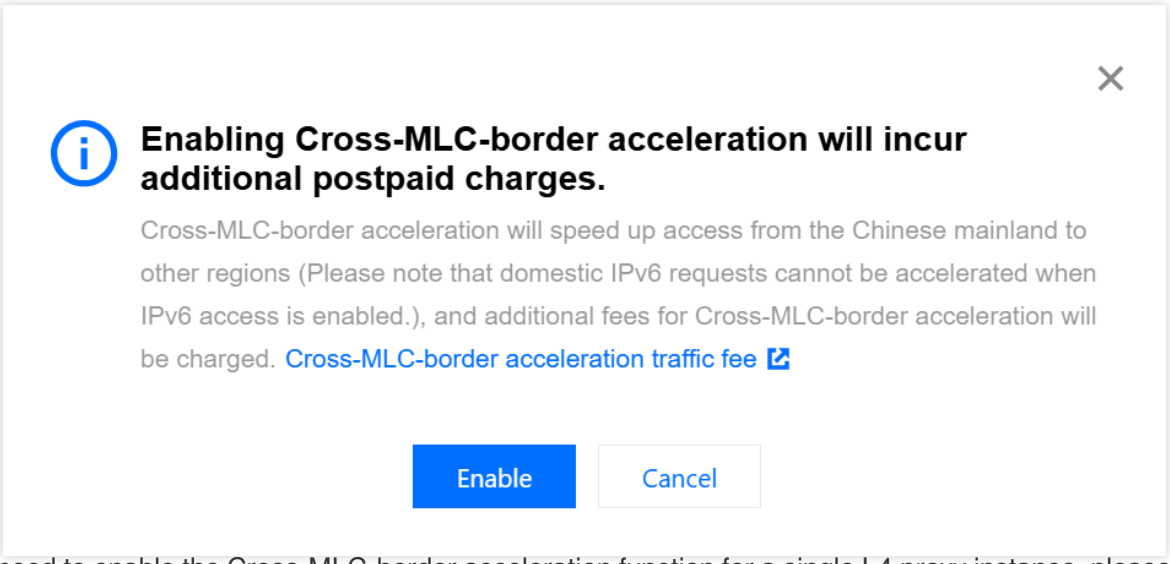
2. If you have also purchased **Exclusive DDoS Protection** , when your domain name is under attack, traffic from the Chinese mainland will prioritize using exclusive DDoS protection resources, and Chinese mainland network optimization will be temporarily disabled; after the attack ends, the Chinese mainland network optimization effect will be restored. During the switch, client connections may be reset.
3. Cross-MLC-border acceleration feature will incur additional Chinese mainland network optimization traffic costs. For details, refer to Chinese mainland network optimization service fees (pay-as-you-go).

Enabling the Cross-MLC-border Acceleration Function

Scenario 1: Configure L7 site-wide acceleration
Scenario 2: Configure a single L4 proxy acceleration
If you need to enable the Cross-MLC-border acceleration function for the entire site, please follow the steps below:
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 Site Global Configuration page. In the right-hand navigation bar, click Network Optimization.
3. On the network optimization page, find the Cross-MLC-border acceleration function configuration card, and click  to enable the Cross-MLC-border acceleration function for the entire site.



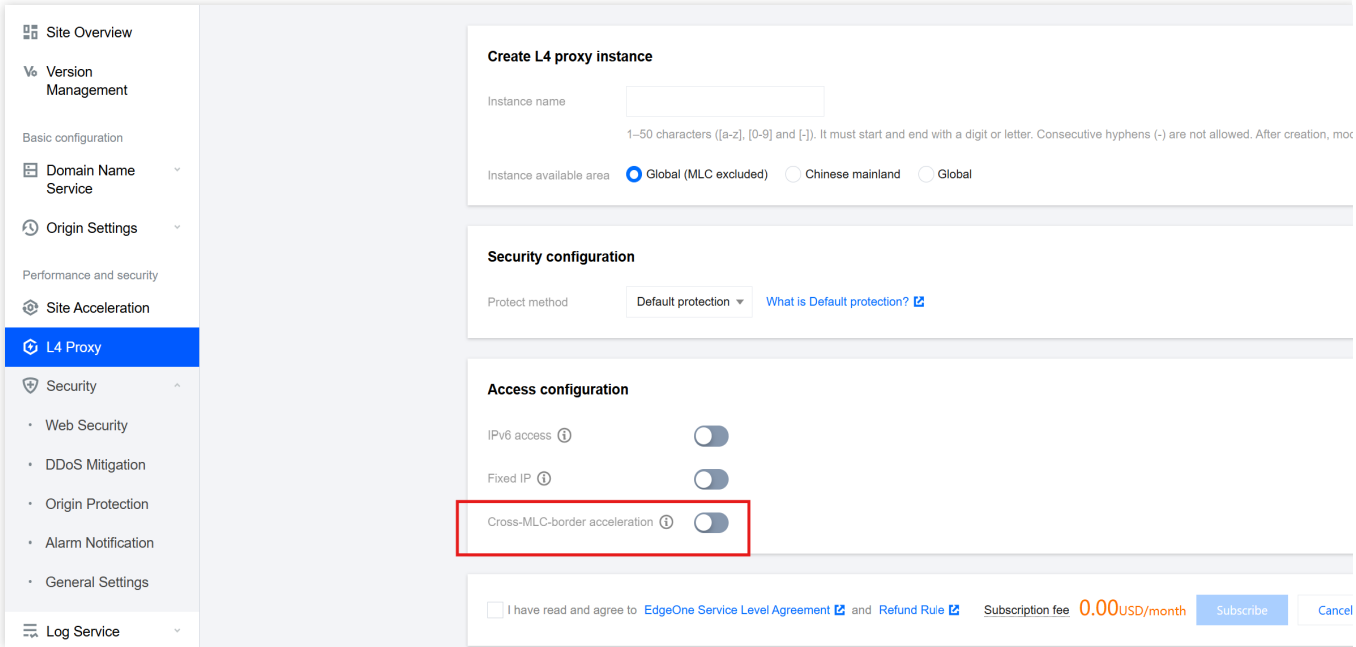
4. In the confirmation window, click **Enable** to complete the configuration.



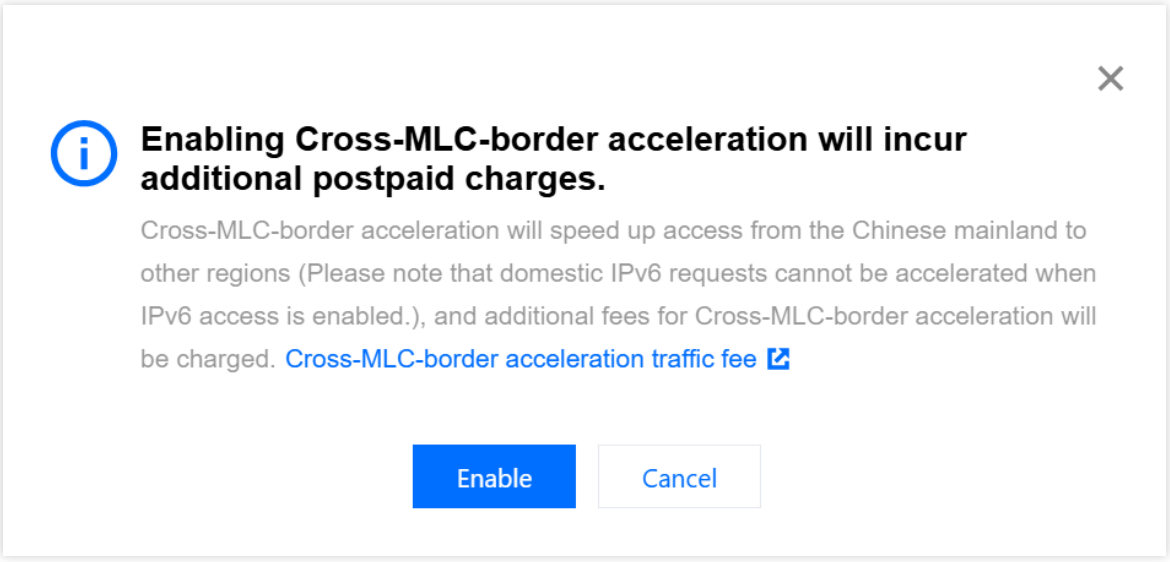
If you need to enable the Cross-MLC-border acceleration function for a single L4 proxy instance, please follow the steps below:

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 **L4 Proxy** and then **Target Instance Name.**
3. Under the target L4 proxy instance, find the Cross-MLC-border acceleration function, and click

to enable the Cross-MLC-border acceleration function for this instance.



4. In the confirmation window, click **Enable** to complete the configuration.



Access Testing

Scenario 1: Configure L7 site-wide acceleration

Scenario 2: Configure a single L4 proxy acceleration

For domains that have enabled the Cross-MLC-border acceleration function, when the customer initiates a visit from the Chinese Mainland, EdgeOne will automatically schedule the access to the Hong Kong access node. You can verify this by checking whether the currently assigned node belongs to Hong Kong, China.

1. You can obtain the IP address of the assigned node by using any of the following methods:

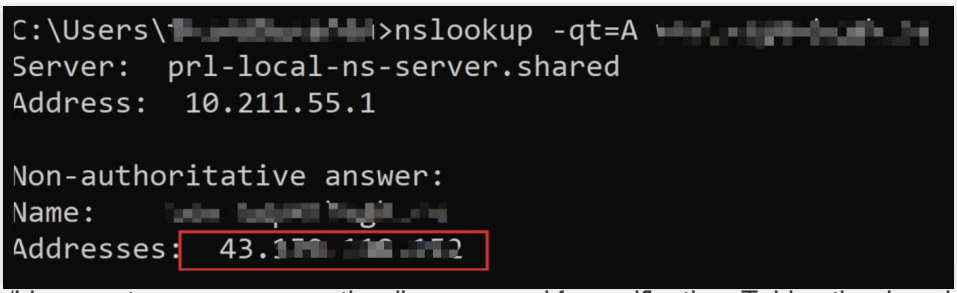
Note： Please ensure that the access test is initiated from the Chinese Mainland since the Cross-MLC-border acceleration function affects the outgoing user requests from the Chinese Mainland.

Windows

Mac/Linux

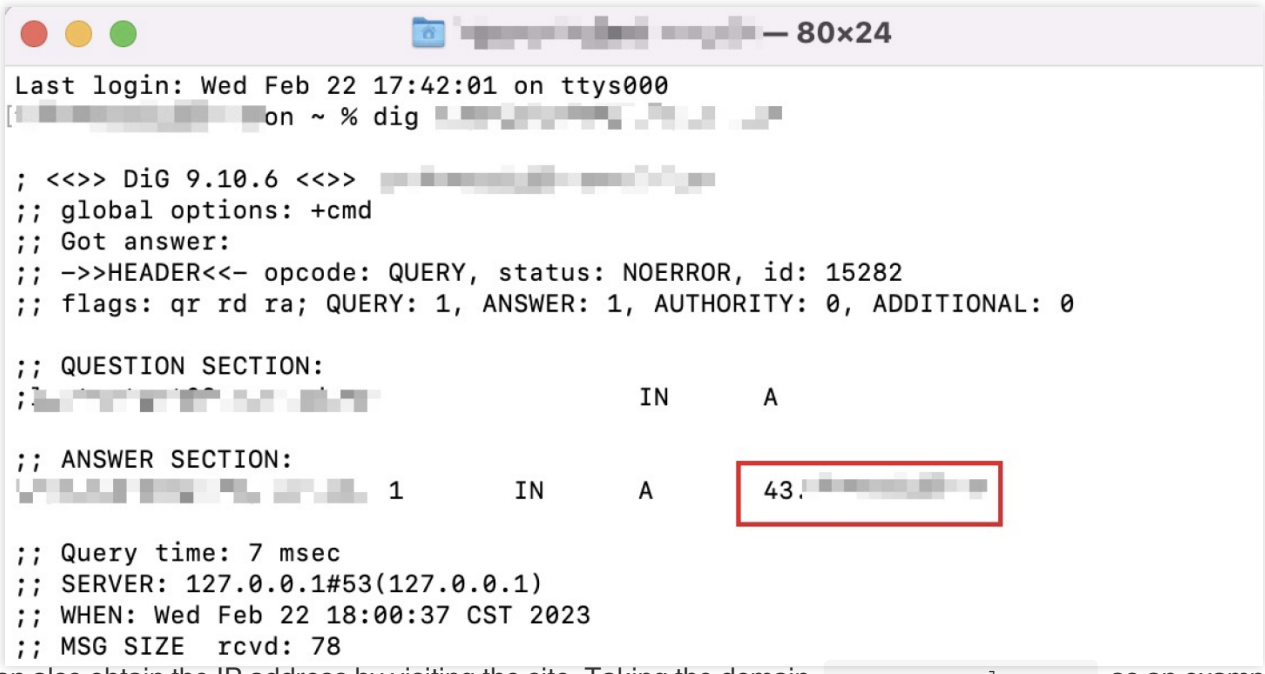
Visit the site

In Windows system, open the command prompt. Taking the domain `www.example.com` as an example, run the `nslookup -qt=A www.example.com` command. Then you can get the IP address of the domain obtained by the A record resolution.

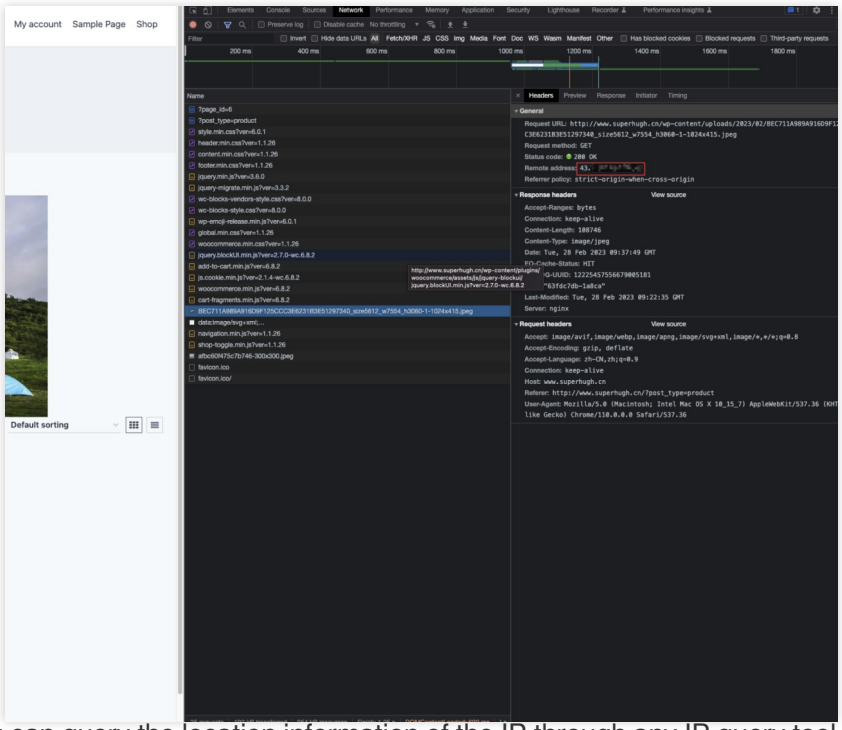


In Mac/Linux system, you can use the dig command for verification. Taking the domain `www.example.com` as an example, run the `dig www.example.com` command in the terminal. Then you can get the IP address of the

domain obtained by the A record resolution.



You can also obtain the IP address by visiting the site. Taking the domain `www.example.com` as an example, you can press F12 in the browser to open the developer tools. Then click any request record, and you can view the IP address that the request points to.



2. You can query the location information of the IP through any IP query tool. If it belongs to Tencent Hong Kong, The Cross-MLC-border acceleration function is effective.

For L4 proxy instances that have enabled the Cross-MLC-border acceleration function, when the customer initiates a visit from the Chinese Mainland, EdgeOne will automatically schedule the access to the Hong Kong access node. You

can verify this by checking whether the currently assigned node belongs to Hong Kong, China.

1. View the L4 proxy instance access domain name. On the site details page, click L4 Proxy. Under the target L4 proxy instances, view the access domain name.

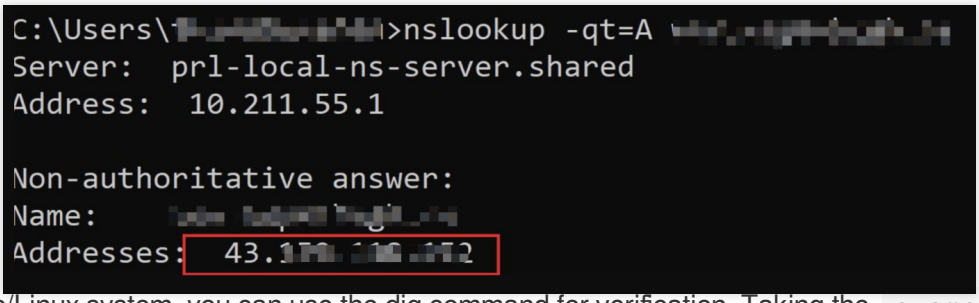
2. You can obtain the IP address of the assigned node by using any of the following methods:

Note： Please ensure that the access test is initiated from the Chinese Mainland since the Cross-MLC-border acceleration function affects the outgoing user requests from the Chinese Mainland,

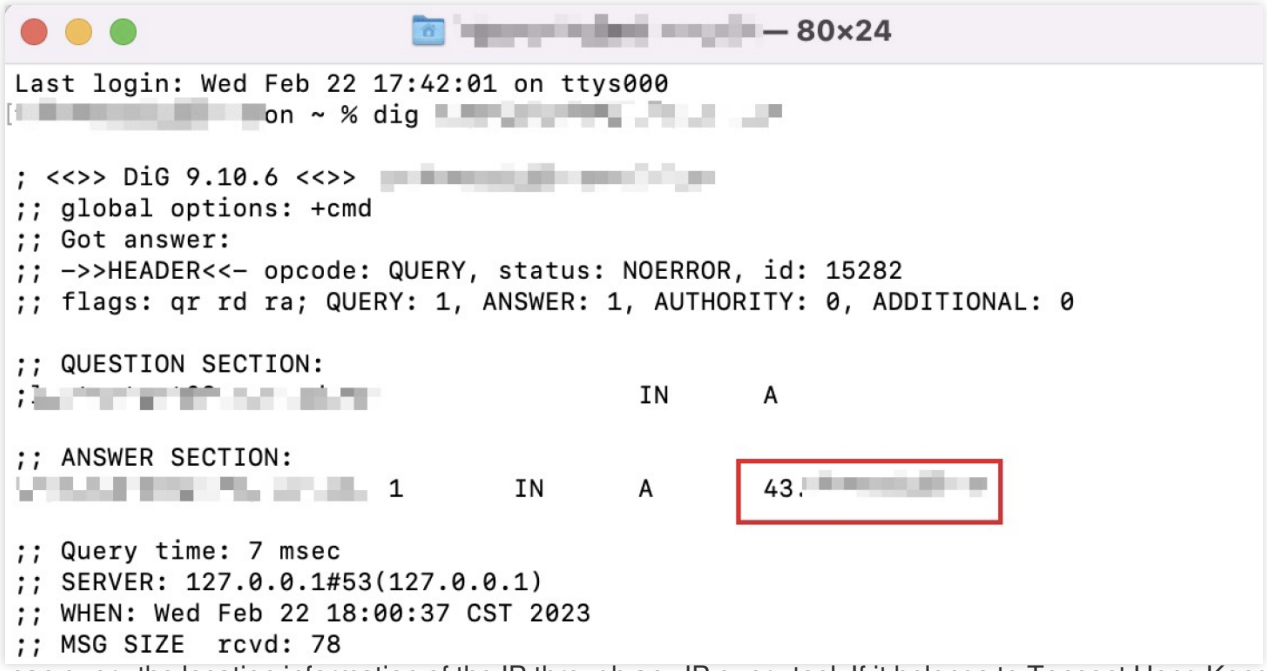
Windows

Mac/Linux

In Windows system, open the command prompt. Taking the domain `example.com.eo.dnse.com` as an example, run the `nslookup -qt=A example.com.eo.dnse.com` command. Then you can get the IP address of the domain obtained by the A record resolution.



In Mac/Linux system, you can use the dig command for verification. Taking the `example.com.eo.dnse.com` as an example, run the `dig example.com.eo.dnse.com` command in the terminal. Then you can get the IP address of the domain obtained by the A record resolution.

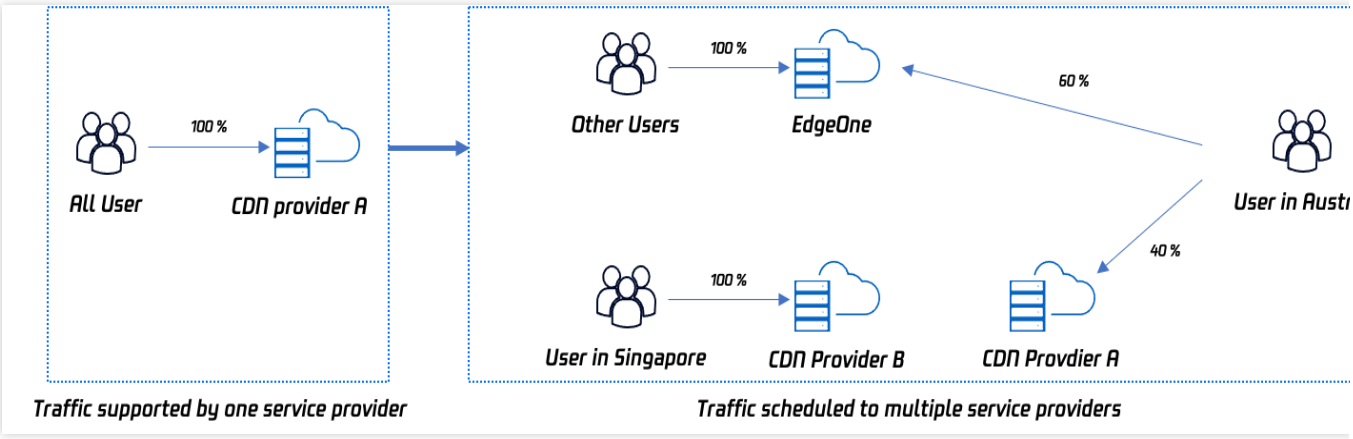


3. You can query the location information of the IP through any IP query tool. If it belongs to Tencent Hong Kong, the Chinese mainland network optimization (international acceleration) function is effective.

Scheduling Traffic Through traffic orchestration to multiple service providers

Last updated：2023-11-08 09:48:05

This article introduces how to use the traffic scheduling feature of EdgeOne Service to help you flexibly allocate the traffic of a domain name to multiple service providers for joint service, disperse risks and achieve high availability for business disaster recovery.



Document Target

This document is expected to take 10 minutes to learn. By studying this document, you can understand:
1. What is traffic scheduling management?
2. How to use traffic scheduling to distribute traffic to multiple service providers for joint service.
3. How to ensure high availability of services through traffic scheduling.

Background Introduction

Websites purchase security acceleration services to improve user access experience and business security, but do not want to schedule all traffic to one service provider. In case of failure, the impact is significant, and traffic needs to be flexibly allocated to multiple service providers for joint service to reduce risks and achieve high availability. The traditional solution is for users to use their own DNS service providers to perform complex configuration pointing for

domain names, such as setting different service providers according to regions, operators, etc. The operation and management are relatively complex. EdgeOne provides traffic scheduling management tools, allowing users to allocate traffic according to countries, provinces, regions, operators, etc., and quickly change and switch services to ensure high availability of business disaster recovery.

Prerequisites

1. Add a site according to the Site Access Guide, purchase the EdgeOne Enterprise plan, and connect the site through CNAME.
2. Add the domain name that needs traffic scheduling switching in the EdgeOne console, and configure it according to the CNAME Access Mode Add Domain Name Guide.

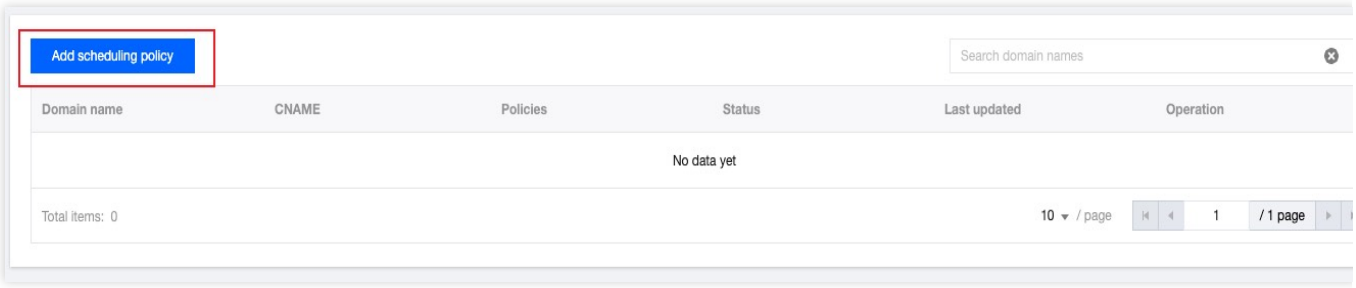
Preset Scenarios

Assume that the domain name a.exampe.com currently uses CDN provider B for all traffic, and consider introducing other providers for joint scheduling. At the same time, when a certain provider encounters problems, traffic scheduling can be switched.
Overall scheduling strategy:
Switch Singapore users to use CDN provider B service.
Australian users use EdgeOne and CDN provider A for joint service, with EdgeOne accounting for 60% and CDN provider A accounting for 40%.
Other regions use the default scheduling and uniformly use EdgeOne service.

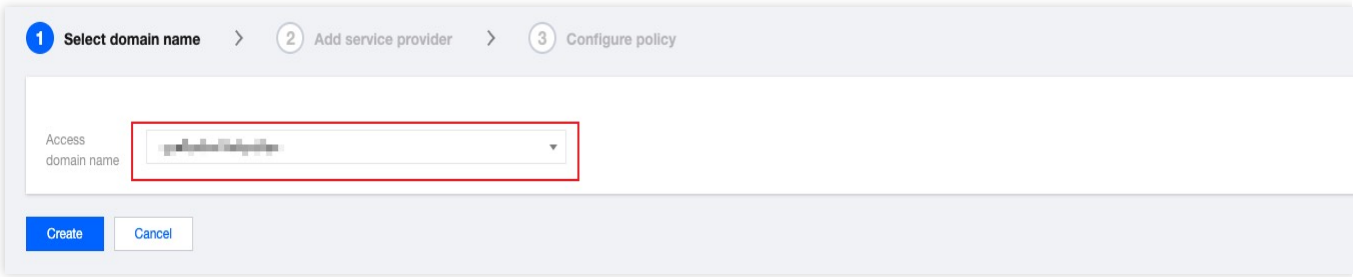
Operation Steps

Step 1: Select the domain name

1. Log in to the EdgeOne console, select Site List from the left navigation, find the site `example.com` where the domain name belongs, and click the site to enter the site management page.
2. After entering the site, click **Domain Name Service > Traffic Scheduling Management** in the menu bar to enter the Traffic Scheduling Management page, and click **Add Scheduling Policy**.

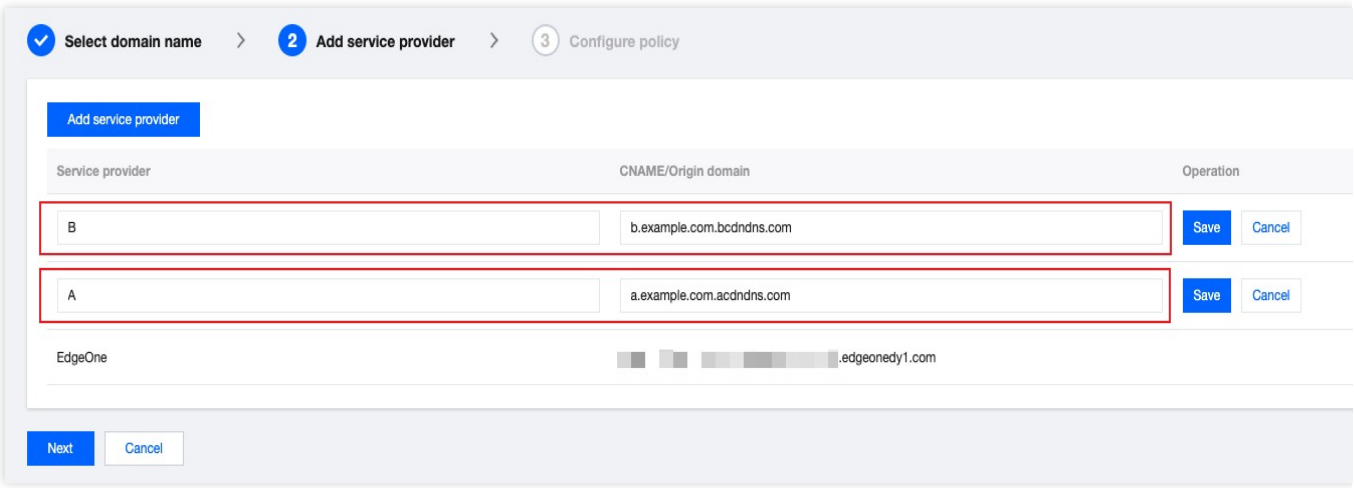


3. On the Traffic Scheduling Management page, click **Add Scheduling Policy**, select `a.example.com`, and click **Create**.



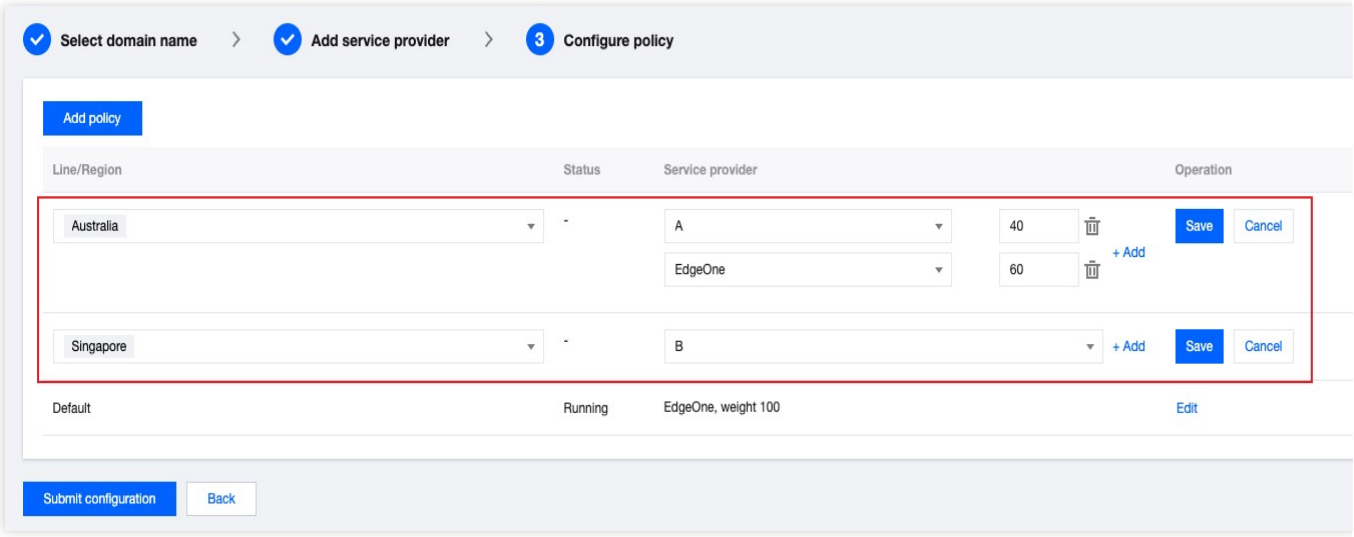
Step 2: Set the policy

1. Add service providers. In this scenario, because it is a multi-provider joint service, the default EdgeOne scheduling CNAME is available, and the CNAME domain names of CDN provider A and CDN provider B can be added separately.



2. Add policy submission configuration, add two policies, and add Chinese mainland and Singapore regions in Line/Region respectively:
Singapore: Select CDN provider B as the service provider.

Australia: Click Add a Service in the service provider section, and select EdgeOne and CDN provider A respectively, with EdgeOne setting a weight of 60 and CDN provider A setting a weight of 40.
Default: By default, others use EdgeOne service.



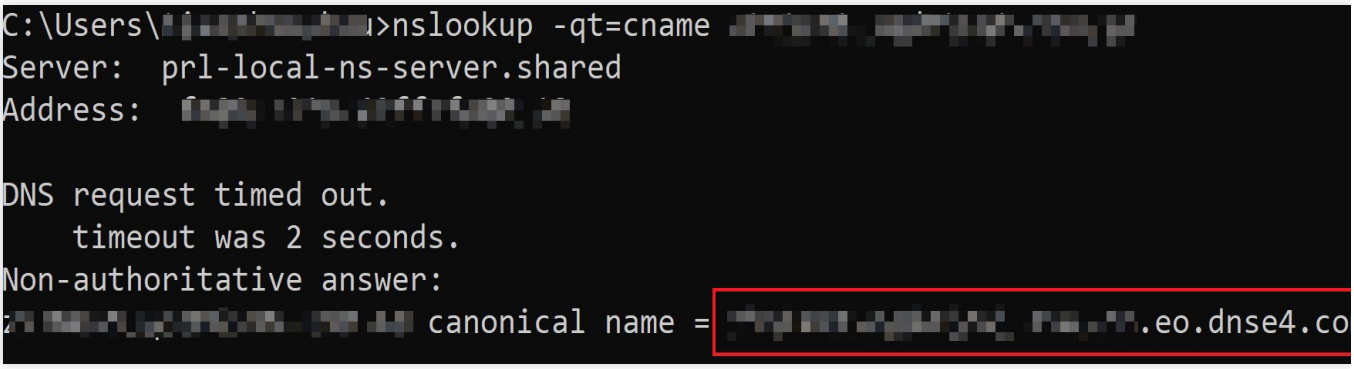
Step 3: Switch resolution

1. After submitting the policy configuration, return to the Traffic Scheduling Management list page. EdgeOne will assign a traffic scheduling CNAME to the domain name, which is consistent with the default CNAME of the domain name.
2. If the domain name resolution has been switched to EdgeOne, no change is required, and the current network policy takes effect immediately. If the domain name resolution has not been switched, you need to go to your DNS service provider to complete the CNAME configuration before the traffic scheduling policy can take effect.

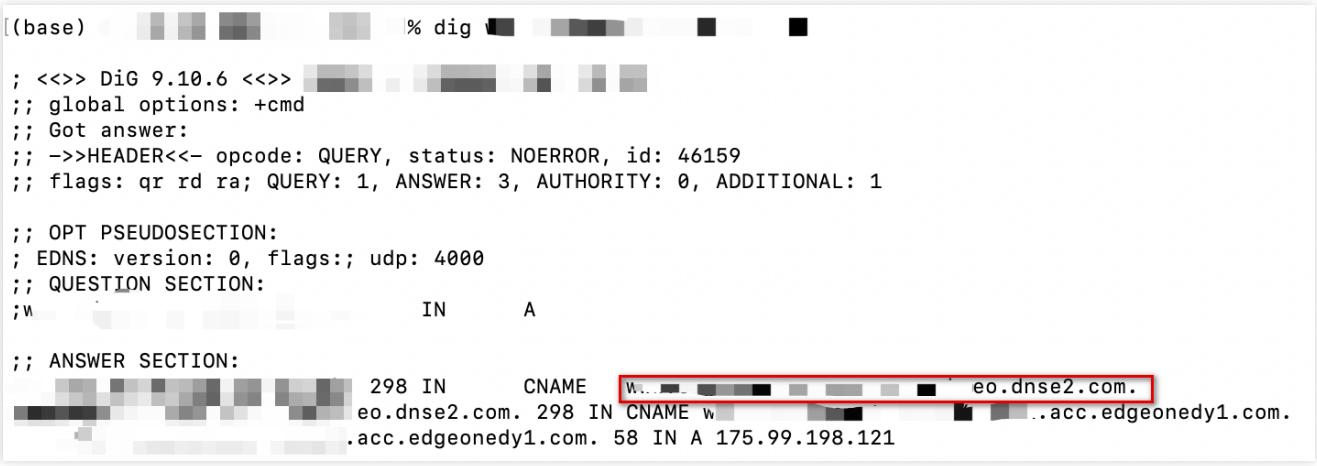
Step 4: Verify Effectiveness

1. DNS resolution effectiveness check

You can use the nslookup or dig command to check the current domain name resolution effectiveness status.
Windows
Mac or Linux
In the Windows system, open the cmd running program, take the domain name `a.example.com` as an example, and judge the effectiveness of the Chinese mainland region. You can run in cmd: `nslookup -qt=cname a.example.com`, and check the CNAME information of the domain name according to the running resolution result. If the CNAME assigned by EdgeOne appears, the traffic switch is successful.



You can use the dig command to verify, take the domain name `a.example.com` as an example, you can run the command in the terminal: `dig a.example.com` , and check the CNAME information of the domain name according to the running resolution result. If the CNAME assigned by EdgeOne appears, the traffic switch is successful.



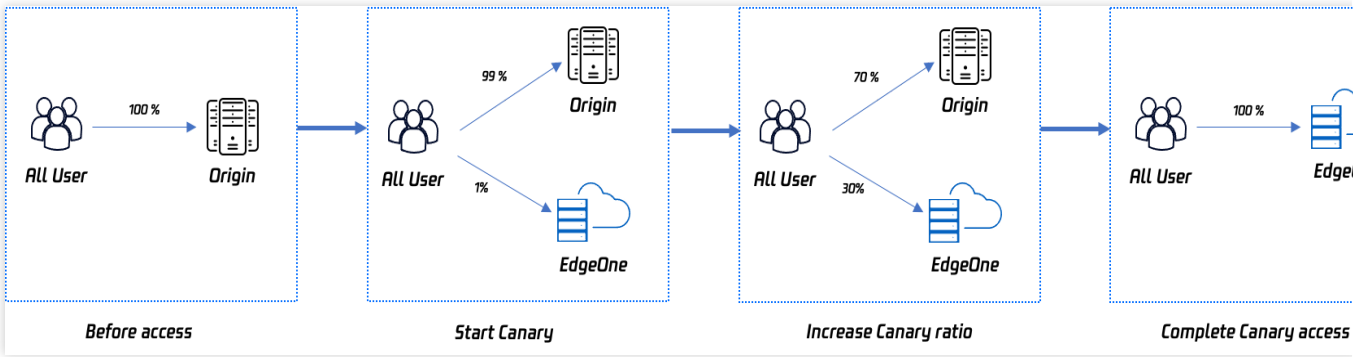
2. Traffic statistics change

Take Singapore as an example, enter the traffic analysis page of site `a.example.com` 's data analysis, add a filter condition host equals `a.example.com` , and check the traffic trend curve change.
For example: The current Singapore bandwidth is 100Mpbs. When Singapore switches to EdgeOne, the bandwidth curve of the EdgeOne console will increase to 100Mbps bandwidth.

Scheduling Traffic to EdgeOne by Performing Canary Switching

Last updated：2023-10-13 14:36:31

This document describes how to perform canary switching to smoothly migrate the business traffic of a domain name from its origin to Tencent Cloud EdgeOne by using the traffic scheduling feature.



Purpose

It may take you 10 minutes to read this document, which helps you:
1. Understand what is traffic scheduling management.
2. Understand how to use the traffic scheduling feature to perform canary switching for traffic migration while guaranteeing high service availability.

Background

After you purchase the Tencent Cloud EdgeOne service, you need to switch the traffic of your website from the origin or other service providers to EdgeOne. A conventional solution requires you to use a tool and access a node for testing and, if the test succeeds, switch the traffic once and for all with one click. This may cause issues in some regions, resulting in availability degradation or bursts of traffic at the origin.

A better solution is to perform canary switching to achieve smooth business migration with guaranteed high service availability. EdgeOne provides the traffic scheduling feature for you to control the canary switching progress by specifying custom traffic migration ratios.

Prerequisites

1. You have added a site, purchased the EdgeOne Enterprise plan, and connected the site to EdgeOne in CNAME access mode. For more information, see Adding Sites.
2. You have added the domain name for canary switching in the EdgeOne console. For more information, see Connecting via CNAME.

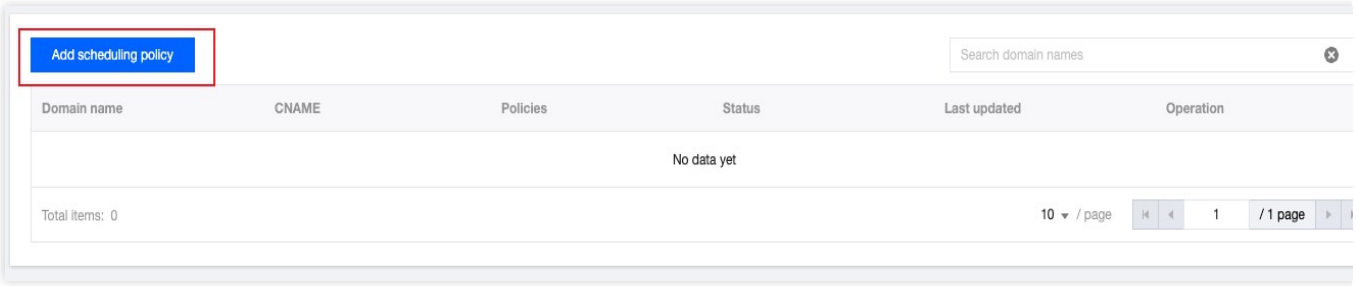
Use Cases

Assume that you want to migrate the traffic of a site, whose domain name is `huidu.example.com` . Currently, the traffic is fully directed to the origin server, whose address is `origin.example.com` .
You plan to switch the traffic to EdgeOne in canary mode by specifying the traffic migration ratio of 1% for the first stage, 30% for the second stage, and 100% for the third stage.

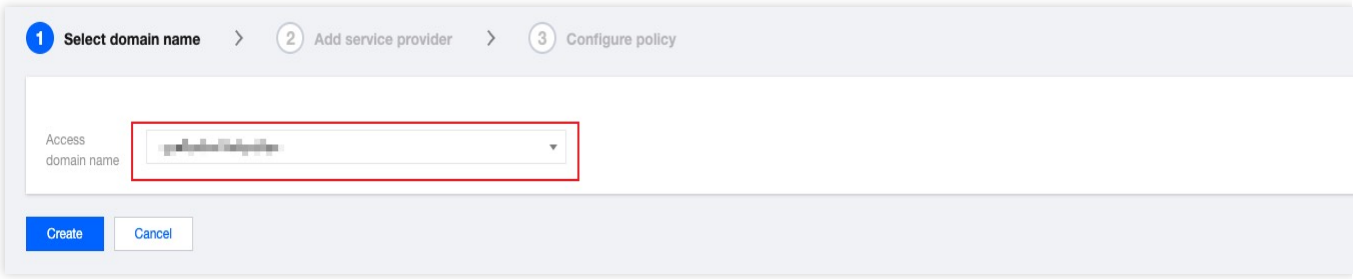
Directions

Step 1. Add an initial canary switching policy

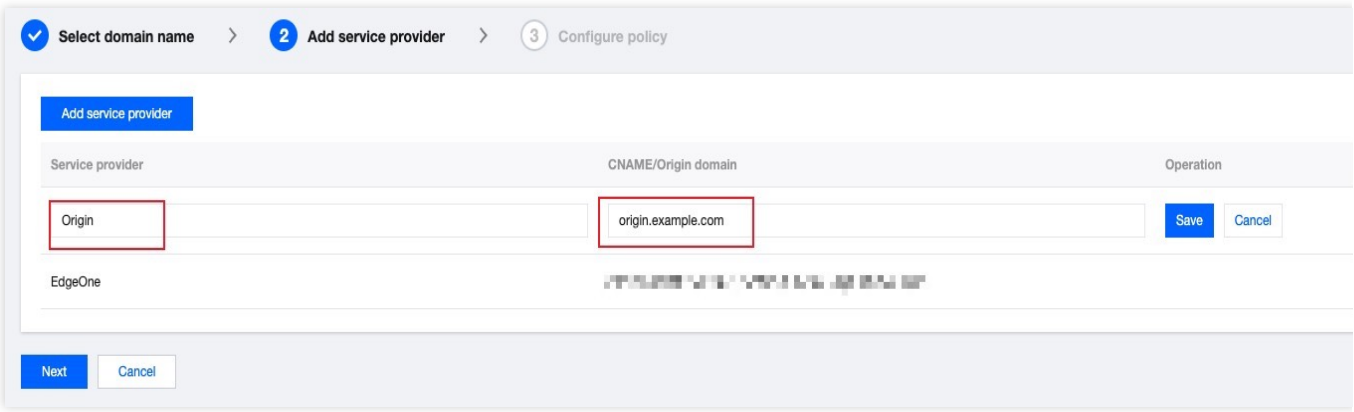
1. Log in to the EdgeOne console and click **Site List** in the left sidebar. In the site list, find the site `example.com` and click the site name.
2. On the site management page, choose **Domain Name Service** > **Traffic Scheduling** in the left sidebar. On the **Traffic Scheduling** page, click **Add scheduling policy**.



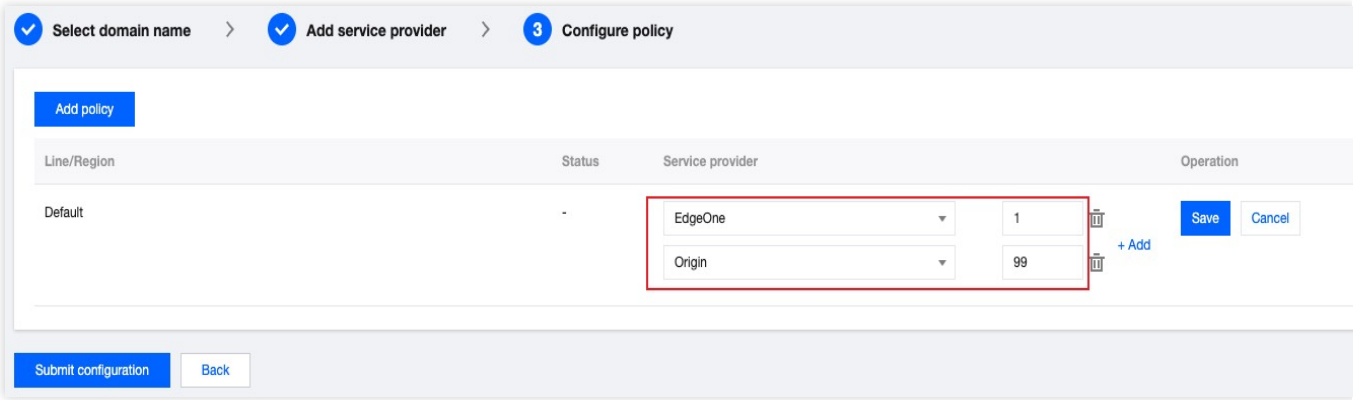
3. In the **Select domain name** step, select `huidu.example.com` from the Access domain name drop-down list and click **Create**.



4. In the **Add service provider** step, specify a custom service provider name, such as `origin domain name`, and enter `origin.example.com` as the origin domain name. This is because the traffic is migrated from the origin in this example. Then, click **Next**.



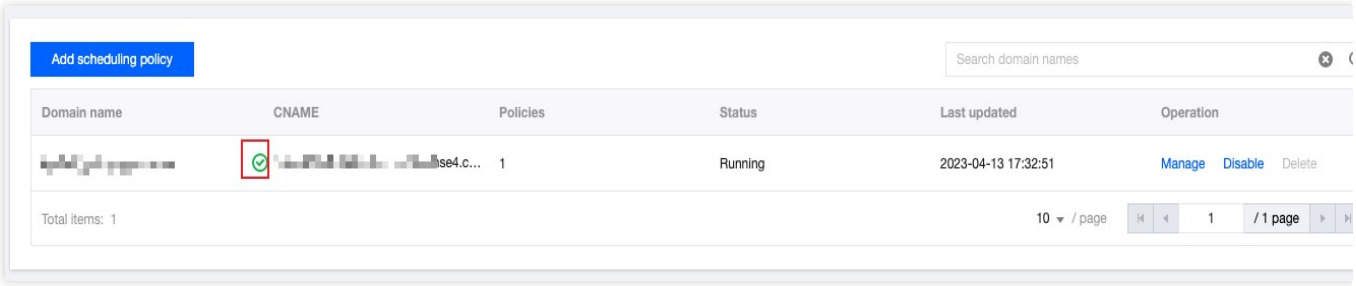
5. In the **Configure policy** step, add an initial canary switching policy and click **Submit**. Set the weight of the service provider `origin domain name` to `99` and that of EdgeOne to `1`. This policy means to switch 1% of traffic from the origin to EdgeOne. You can increase the traffic migration ratio later if the service remains stable.



Step 2. Start canary switching

1. Configuring DNS

After you added the policy, EdgeOne assigns a CNAME record for traffic scheduling to the domain name. The assigned CNAME record is the same as the default CNAME record of the domain name. You need to configure the CNAME record at your DNS service provider to activate the traffic scheduling policy. For more information, see Step 4 in Connecting via CNAME.



2. Verifying the switching result

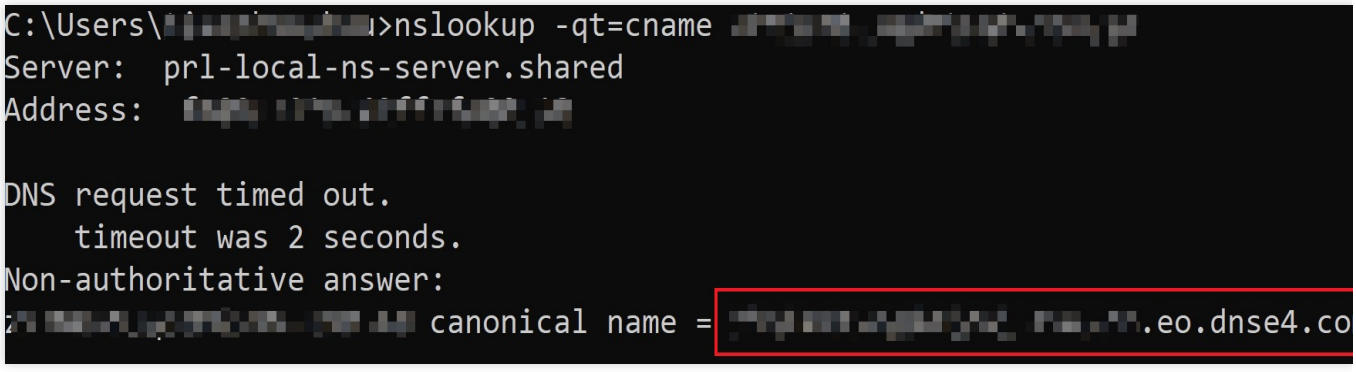
You can run the `nslookup` or `dig` command to check the switching result.

Windows

macOS or Linux

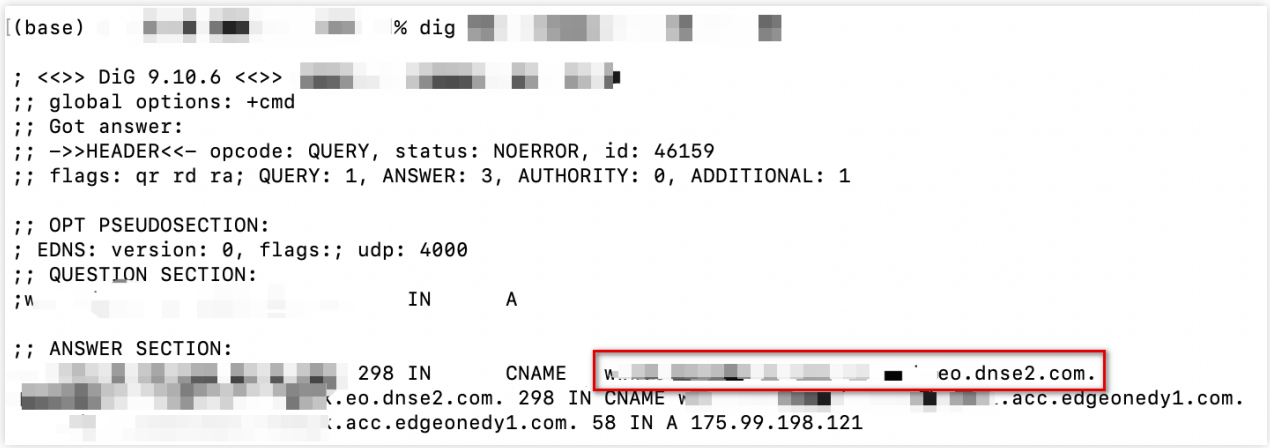
Open the command prompt and run `nslookup -qt=cname huidu.example.com` . Then, check the ratio of the CNAME addresses in the DNS result.

In this example, you have specified the traffic migration ratio of 1%. Therefore, if the traffic switching is successful, about 1% of the returned CNAME addresses are provided by EdgeOne. You can run the command several times.



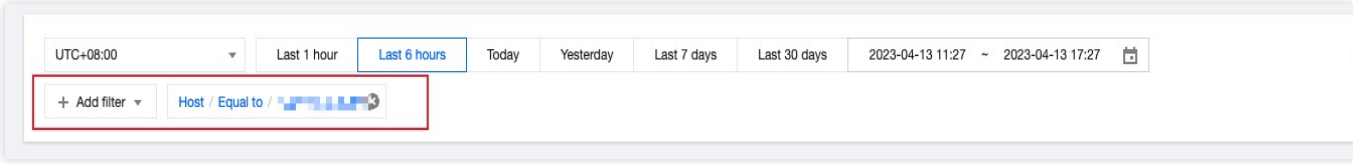
Open the terminal and run `dig huidu.example.com` . Then, check the ratio of the CNAME addresses in the DNS result.

In this example, you have specified the traffic migration ratio of 1%. Therefore, if the traffic switching is successful, about 1% of the returned CNAME addresses are provided by EdgeOne. You can run the command several times.



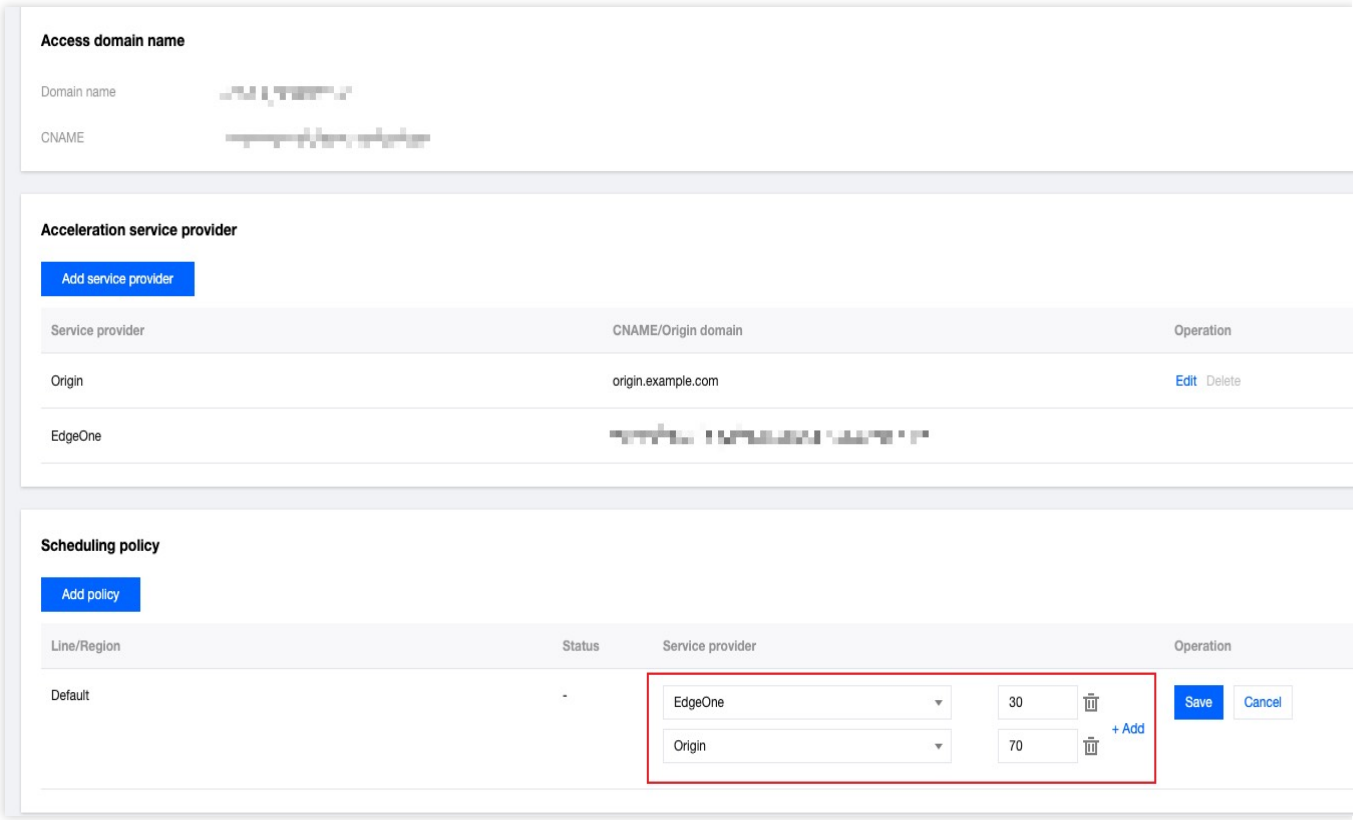
3. Viewing traffic changes

Choose **Data Analysis** > **Traffic Analysis** in the left sidebar and filter the traffic by setting the filter to `Host / Equal to / huidu.example.com`. Then, view the changes of the traffic trend curves. For example, if the total bandwidth is 100 Mpbs and 1% of the traffic is switched to EdgeOne, the bandwidth curve will raise to 1 Mbps.



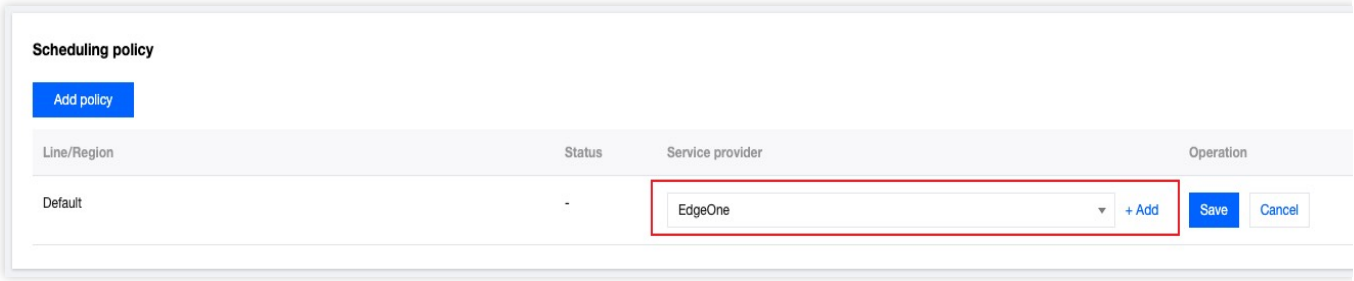
Step 3. Increase the traffic migration ratio

To increase the traffic migration ratio to 30%, go to the **Traffic Scheduling** page, find `huidu.example.com`, and click **Manage** in the **Operation** column. On the page that appears, change the weight of EdgeOne to 30 and that of the origin to 70, and click **Save**. The policy will take effect after the DNS cache expires. Then, verify the switching result. For more information, see 2. Verifying the switching result in Step 2.



Step 4. Switch the traffic in full

Perform the following operations to increase the traffic migration ratio to 100% and fully switch the traffic to EdgeOne.
1. Delete the service provider `origin domain name` and click **Save**. The policy will take effect after the DNS cache expires. Then, verify the switching result. For more information, see 2. Verifying the switching result in Step 2.



2. You can disable and delete the traffic scheduling policy later if the service remains stable after 100% canary switching. At this point, disabling or deleting the policy has no impact on the service, and the traffic is fully managed by EdgeOne.

Relevant Documentation

Origin-pull Based On User IP/geolocation EdgeOne Implementation of Session Persistence Based on Client IP Addresses

Last updated：2025-01-14 14:22:21

It will take you about **10** minutes to study this document. By studying this document, you can understand:

1. **What is session persistence based on Client IP address? Why is it necessary?**
2. **Use cases for session persistence based on client IP addresses.**
3. **Technical architecture and principle explanation based on Client IP address.**
4. **The specific steps for implementing session persistence based on client IP addresses with EdgeOne edge functions and rule engines.**

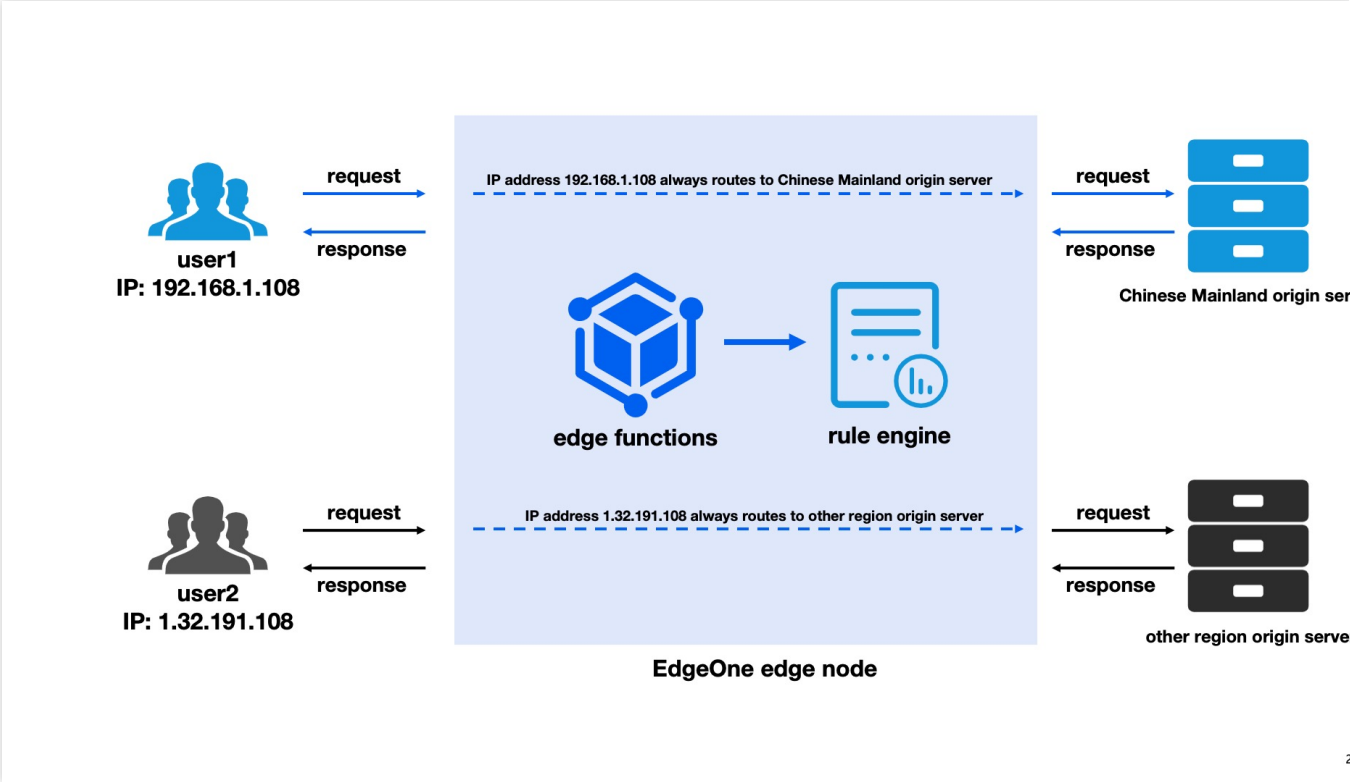
Background

With the rapid development of the Internet, enterprises are constantly expanding their businesses and deepening user experience. A single origin server gradually fails to meet the needs of handling a large number of concurrent requests. To enhance the availability and scalability of services, enterprises are beginning to adopt load balancing technology to distribute user requests to multiple backend origin servers for processing. However, in the early stages of business development, due to the relatively small number of users and relatively simple session management, session persistence based on client IP addresses is usually not required. As the business further develops, especially in the following scenarios, the demand for session persistence based on client IP addresses becomes particularly urgent:
User login status persistence: In applications requiring user login, such as e-commerce websites and online banking, a session is created on the origin server after the user logs in to record user login status, shopping cart information, order details, and other information. If the user is assigned to different backend origin servers during browsing, the user needs to log in again due to the loss of session information, severely affecting user experience.
Businesses with high data consistency requirements: In businesses with strict data consistency requirements, such as financial transactions and online payments, distributing sessions to different origin servers may lead to data inconsistency or loss, causing significant losses for both users and enterprises.
EdgeOne introduces session persistence based on Client IP address in the above context. This is achieved by using EdgeOne's Edge Functions and Rule Engine to ensure that requests from the same Client IP address are always forwarded to the same backend or.

Use Cases

By identifying the client's IP address to ensure that requests from the same client are directed to the same origin server, this feature is applicable to the following business scenarios:
Financial services: Applications like online banking and stock trading need to ensure that all requests from a user throughout the transaction are processed by the same origin server to maintain transaction security and consistency.
E-commerce websites: After a user logs in, all requests from the user are routed to the same origin server to maintain shopping cart information, user preference settings, and login status.

Technology Architecture



The specific principle of session persistence based on Client IP address is that the Edge Function maps the client to different origin servers using a hash algorithm based on the client's IP address (e.g., IP: 192.168.1.108 in the above figure). The Rule Engine retrieves the custom origin request header in the Edge Function and ensures that the same client always returns to the same origin server based on the value of the request header. This achieves consistency from the client to a specific origin server. This solution not only enhances user experience but also ensures the accuracy of business.

Scenario

Assume that you are a technical lead of a global application service, and you have integrated your site domain name `example.com` into EdgeOne. You expect to route requests according to the hash value of the user's IP address to

the corresponding origin server, ensuring that regardless of the user's location, the same user's requests are always routed to the same origin server. This helps optimize cache efficiency, simplify session management, implement load balancing, deliver personalized services, and ensure data processing's legal compliance.
In this scenario, you are dealing with millions of different users, whose requests need to be evenly distributed to the **Chinese mainland origin server group** and **Singapore origin server group**. At the same time, you expect requests from the same IP address to be always routed to the same origin server to achieve a consistent user experience and efficient resource usage. In this example, IP addresses that need to be forwarded to the Chinese mainland origin server group will have an origin-pull request header `X-Forwarded-For-Origin:originGroup1` added by the edge function, and IP addresses that need to be forwarded to the Singapore origin server group will have an origin-pull request header `X-Forwarded-For-Origin:originGroup2` added by the edge function.

Directions

Step 1: Connecting to EdgeOne

Refer to Quick Start to complete site connection and domain name connection.

Step 2: Creating and Configuring the Edge Function

1. Log in to the EdgeOne console, select the site to be configured from the **Site List**, and enter the site management submenu.
2. In the left navigation bar, click **Edge Functions** > **Function Management**.
3. On the Function Management page, click **Create function**.
4. On the template selection page, select **Create Hello World**, and then click **Next**.
5. On the function creation page, enter the function name, description, and code. Below is sample code for session persistence based on client IP addresses:

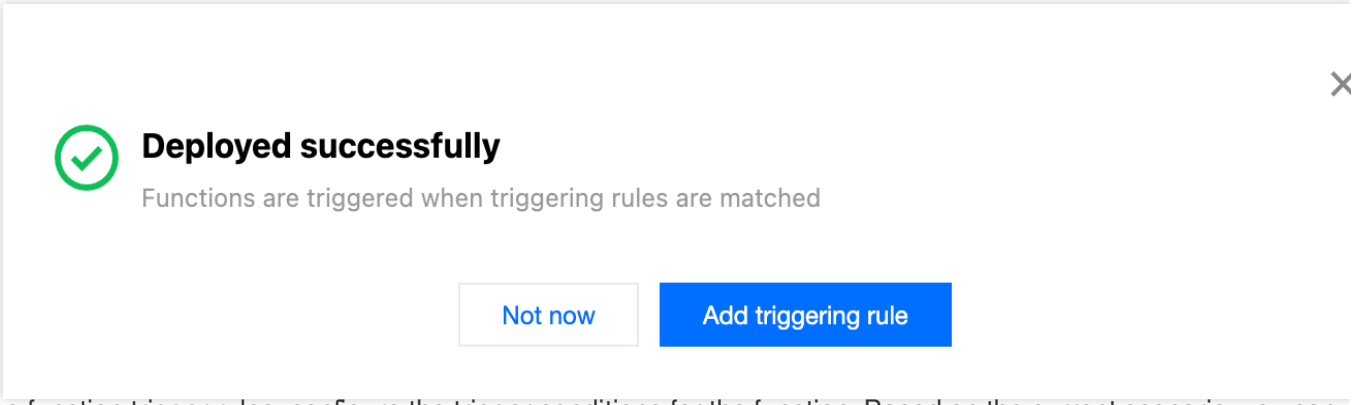
```
// Based on client IP addresses, return the clients to different origin server grou
const ORIGIN_GROUPS = ["originGroup1", "originGroup2"];

// Define the number of virtual nodes. If there are many origin server groups (ORIG
const VIRTUAL_NODES_PER_GROUP = 15;
const ORIGIN_HEADER_NAME = 'X-Forwarded-For-Origin';
let virtualNodesHashesCache = null;

// Define global variables to track the function invocation count.
addEventListener("fetch", (event) => {
  handleRequest(event.request);
});

async function handleRequest(request) {
```

```
  // Obtain the client IP address through the EO-Client-IP header.
  const ip = request.headers.get("EO-Client-IP") || "";

  // If there are no hash values for virtual nodes in the cache, generate the hash
  if (!virtualNodesHashesCache) {
    virtualNodesHashesCache = await generateVirtualNodesHashes();
  }

  const group = await findSourceGroupForIp(
    ip,
    virtualNodesHashesCache.hashes,
    virtualNodesHashesCache.mapping
  );
  console.log(`Group: ${group}`);

  request.headers.set(ORIGIN_HEADER_NAME, group)

  return;
}

// Generate virtual nodes' hash values.
async function generateVirtualNodesHashes() {
  const virtualNodesHashes = {};

  for (let group in ORIGIN_GROUPS) {
    for (let i = 0; i < VIRTUAL_NODES_PER_GROUP; i++) {
      const virtualNodeIdentifier = `${group}-VN${i}`;
      const hash = await md5(virtualNodeIdentifier);
      if (!virtualNodesHashes[hash]) {
        virtualNodesHashes[hash] = group;
      }
    }
  }

  const hashes = Object.keys(virtualNodesHashes).sort();

  return { hashes, mapping: virtualNodesHashes }; // Return the sorted array of has
}

// Map the client IP address to the virtual node and find the corresponding origin
async function findSourceGroupForIp(ip, hashes, mapping) {

  // Use the MD5 function to calculate the hash value of the IP address.
  const ipHash = await md5(ip);
  let closestHash = hashes.find((hash) => hash > ipHash) || hashes[0];

  // Based on the found hash value of the virtual node, obtain the corresponding or
```

```
  const selectedGroupName = mapping[closestHash];
  const selectedGroupIPs = ORIGIN_GROUPS[selectedGroupName];

  // Print logs that display the client IP address, hash value, closest virtual nod
  console.log(
    `IP: ${ip}, Hash: ${ipHash}, Closest Hash: ${closestHash}, Selected Group: ${se
  );

  // Return the selected origin server group name.
  return selectedGroupIPs;
}

function bufferToHex(arr) {
  return Array.prototype.map
    .call(arr, (x) => (x >= 16 ? x.toString(16) : "0" + x.toString(16)))
    .join("");
}

async function md5(text) {
  const buffer = await crypto.subtle.digest("MD5", TextEncoder().encode(text));
  return bufferToHex(new Uint8Array(buffer));
}
```

Step 3: Configuring and Deploying Trigger Rules for the Edge Function

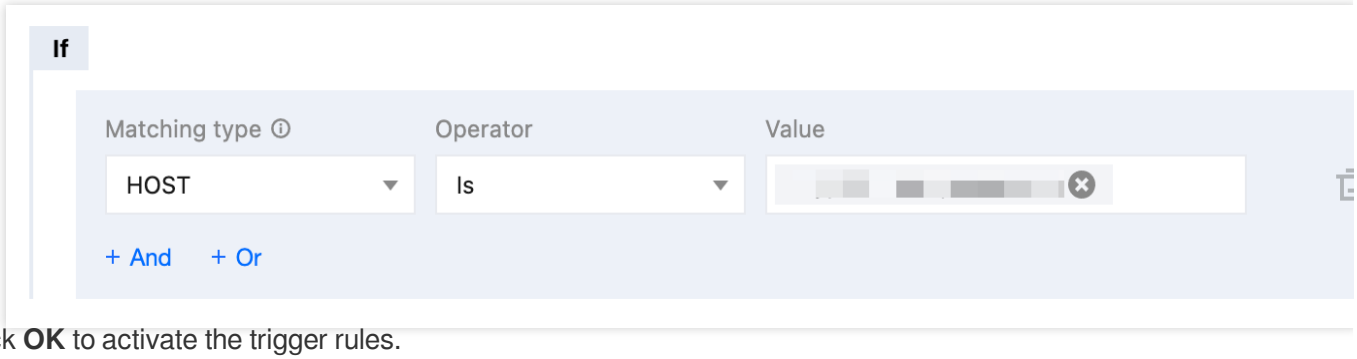
1. After editing the function, click **Create and deploy**. Once the function is deployed, you can directly click **Add triggering rule** to configure the trigger rules for the function.



2. In the function trigger rules, configure the trigger conditions for the function. Based on the current scenario, you can configure multiple trigger conditions using AND logic.

Here, configure the request HOST as `example.com` .

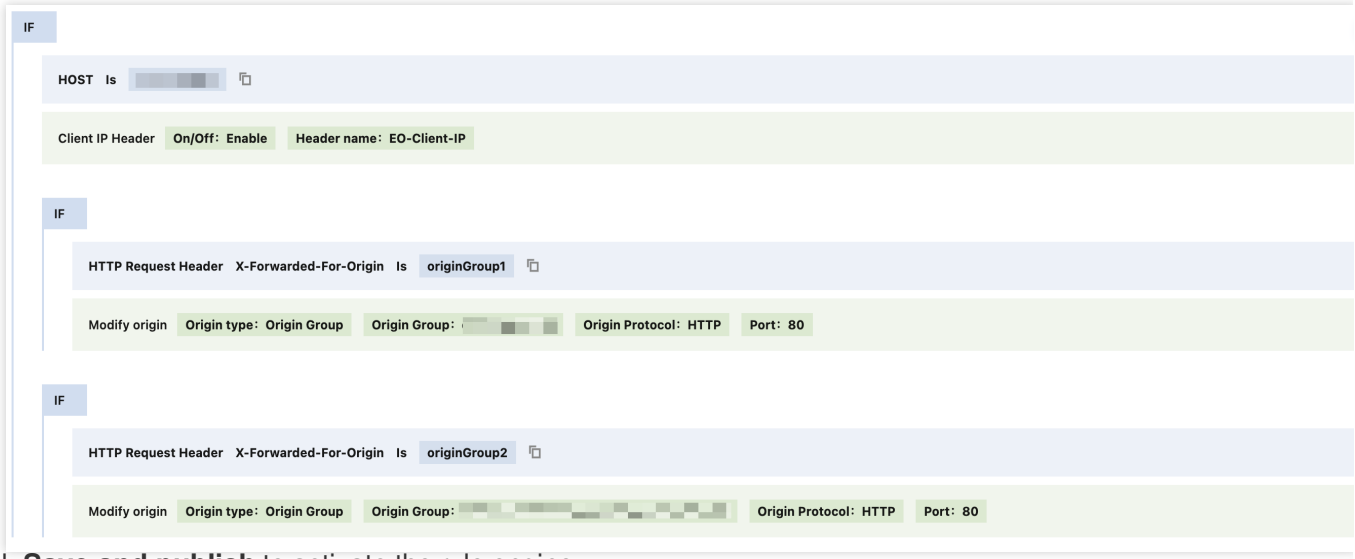
When the request URL meets the above conditions, the edge function in Step 2 will be triggered, implementing session persistence based on client IP addresses.



3. Click **OK** to activate the trigger rules.

Step 4: Configuring the Rule Engine

1. Log in to the EdgeOne console, select the site to be configured from the **Site List**, and enter the site management submenu.
2. In the left navigation bar, click **Site Acceleration** to enter the global configuration page for the site. Then, click the **Rule Engine** tab.
3. On the Rule Engine page, click **Create rule**, and then select **Add blank rule**.
4. On the rule editing page, select HOST as the matching type to match requests for a specific domain name.
Here, configure the request HOST as `example.com` .
5. On the rule editing page, enable **Client IP Header** by referring to Obtaining Client IP Address.
Here, configure the header name as `EO-Client-IP` .
6. On the rule editing page, click **+IF**, and based on the request header values in edge function code, configure different origin server groups.
Here, configure that when the HTTP request header `X-Forwarded-For-Origin` equals `originGroup1` , the request will be forwarded to the origin server group in the Chinese mainland for processing; when the HTTP request header `X-Forwarded-For-Origin` equals `originGroup2` , the request will be forwarded to the origin server group in Singapore for processing.



7. Click **Save and publish** to activate the rule engine.

Step 5: Verifying the Deployment Effect

After testing, this example demonstrated good load balancing capabilities, with the load balancing proportion fluctuating around 50%, and it effectively maintained user session consistency, proving that the deployment effect met expectations.

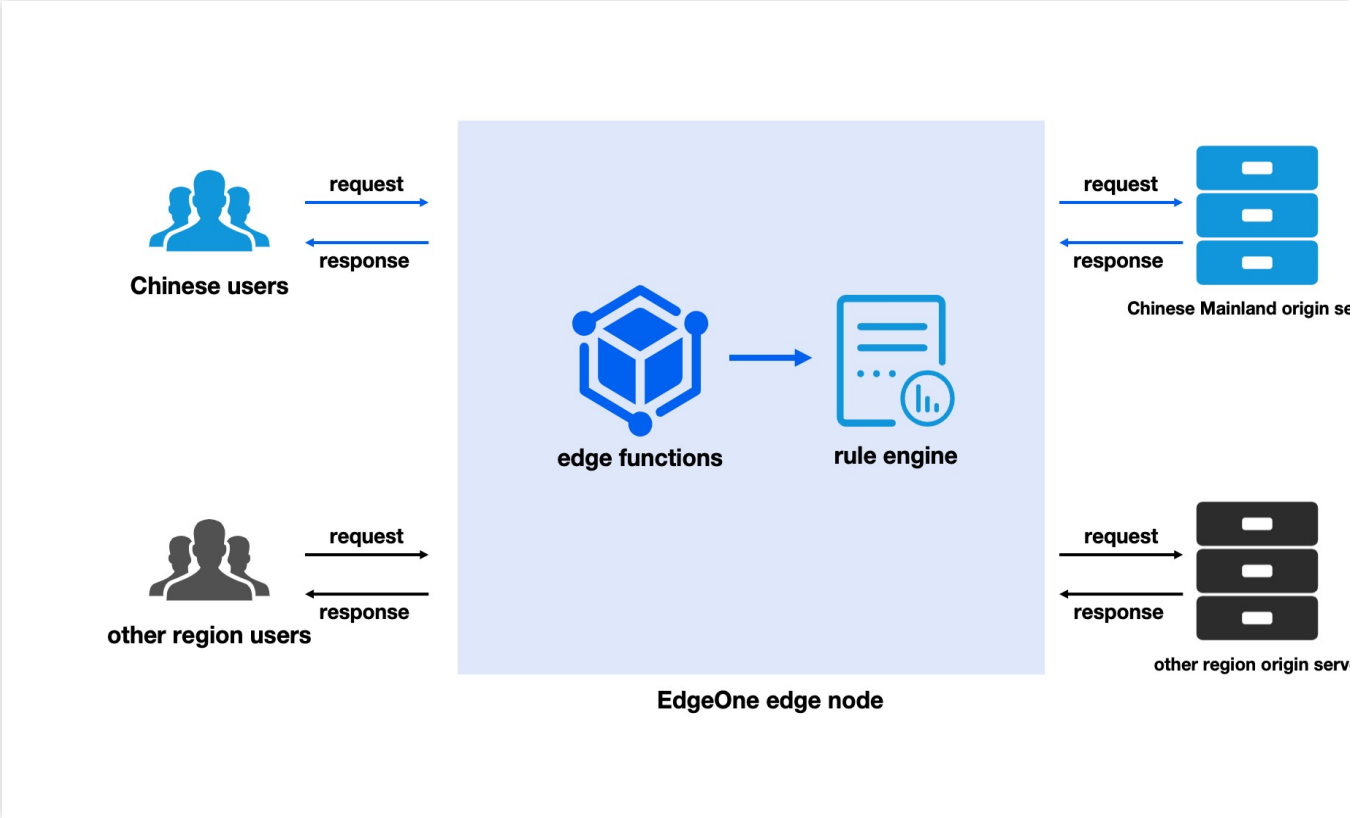
EdgeOne Implementation of Origin-Pull Based on Client's Geo Location

Last updated：2024-12-19 11:18:09

It will take you about **10** minutes to study this document. By studying this document, you can understand:

1. **Why is origin-pull based on client's geo location necessary?**
2. **Use cases for origin-pull based on client's geo location.**
3. **The specific steps for implementing origin-pull based on client's geo location with EdgeOne edge functions and rule engines.**

Background



In today's global business environment, enterprises must provide services that transcend geographic boundaries, ensuring a consistent experience for users regardless of their location. Global service demands require enterprises to respond quickly to the needs of users in different regions, providing personalized content and services. Additionally, with the increasing stringency of data protection regulations worldwide, such as the European Union's General Data Protection Regulation (GDPR), enterprises must ensure compliance when handling user data to avoid legal risks and reputational damage.

Through EdgeOne edge functions and rule engines, a solution is implemented that routes users' requests to the nearest specified origin server based on their geo locations. This can address the aforementioned challenges by analyzing users' geo locations and network requests, and routing the requests to the specified optimal origin server. Specifically, the edge function defines the origin-pull request header based on the client's geo location, and the rule engine uses this header to route the request to the specified origin server. This solution not only improves response speed and performance but also ensures compliance with data processing regulations. No matter where the user is in the world, they can enjoy fast and regulation-compliant service experience, helping your enterprise maintain competitiveness in the global market.

Use Cases

EdgeOne distributes requests from global users' clients to the specified origin server based on their geo locations. The following are specific application scenarios:
Enterprise internationalization: Enterprises can provide tailored services globally. For example, in the financial service sector, it ensures that transaction requests are quickly and accurately routed to the nearest server, reducing latency, and simultaneously provides region-specific investment recommendations and market analysis to meet the needs of users in different regions. Additionally, cross-border e-commerce platforms can locate users' geo locations to offer localized products and services, optimizing inventory and logistics policies to enhance user experience and operational efficiency.
Data privacy compliance: As data protection regulations become increasingly stringent, enterprises need to ensure their data processing activities comply with regulations corresponding to users' geo locations. Enterprises can route data requests to servers that comply with local data protection regulations, ensuring data compliance. This not only helps enterprises avoid legal risks but also respects user privacy, building user trust and laying a solid foundation for the enterprise's sustainable development.

Scenario

Assume that you are a technical lead of a global e-commerce platform and you have integrated your site domain name `example.com` into EdgeOne. Your goal is to optimize the website's access speed and user experience, ensuring that global users can quickly access website content. To achieve this goal, you plan to dynamically route requests to the nearest origin server based on users' geo locations while ensuring that data processing activities comply with data protection regulations in the users' respective regions.
In this scenario, you have set two client regions and two corresponding origin server groups:
Chinese mainland client: For users from the Chinese mainland, you intend to route their requests to **the origin server group located in the Chinese mainland**. This ensures that the data is processed locally, reducing data transmission latency and improving access speed. This also helps ensure that data processing activities comply with

Chinese mainland data protection regulations, protecting user privacy and avoiding legal risks. Requests routed to the Chinese mainland origin server will have an origin-pull request header `X-Forwarded-For-Origin:cn` added by the edge function.

Singapore client: For users from Singapore (representing regions outside the Chinese mainland), you intend to route their requests to **the origin server group located in Singapore**. By leveraging the geographic advantage to reduce latency, user experience is enhanced while ensuring data processing activities comply with Singapore data protection regulations, maintaining user data privacy security. Requests routed to the Singapore origin server will have an origin-pull request header `X-Forwarded-For-Origin:sg` added by the edge function.

Directions

Step 1: Connecting to EdgeOne

Refer to Quick Start to complete site connection and domain name connection.

Step 2: Creating and Configuring the Edge Function

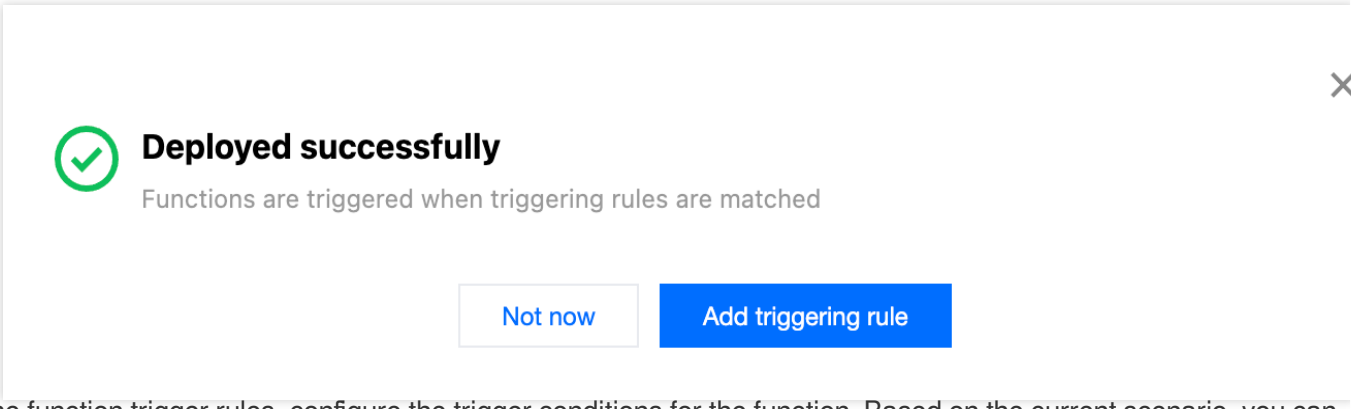
1. Log in to the EdgeOne console, select the site to be configured from the **Site List**, and enter the site management submenu.
2. In the left navigation bar, click **Edge Functions** > **Function Management**.
3. On the Function Management page, click **Create function**.
4. On the template selection page, select **Create Hello World**, and then click **Next**.
5. On the function creation page, enter the function name, description, and code. Below is sample code for origin-pull based on client's geo location:

```
// Domain name request region and origin server group mapping table
const ROUTE_CLIENT_ORIGIN_MAP = {
  'example.com': {
    CN: 'cn',
    _DEFAULT_: 'sg',
  },
};

// Define the HTTP header name used to identify the original client region.
const ORIGIN_HEADER_NAME = 'X-Forwarded-For-Origin';
// Define the identifier of regions outside the Chinese mainland.
const OVERSEAS_AREA = '!CN';
// Define the identifier of the default region.
const DEFAULT_AREA = '_DEFAULT_';

addEventListener('fetch', (event) => {
  event.respondWith(handleEvent(event));
});
```

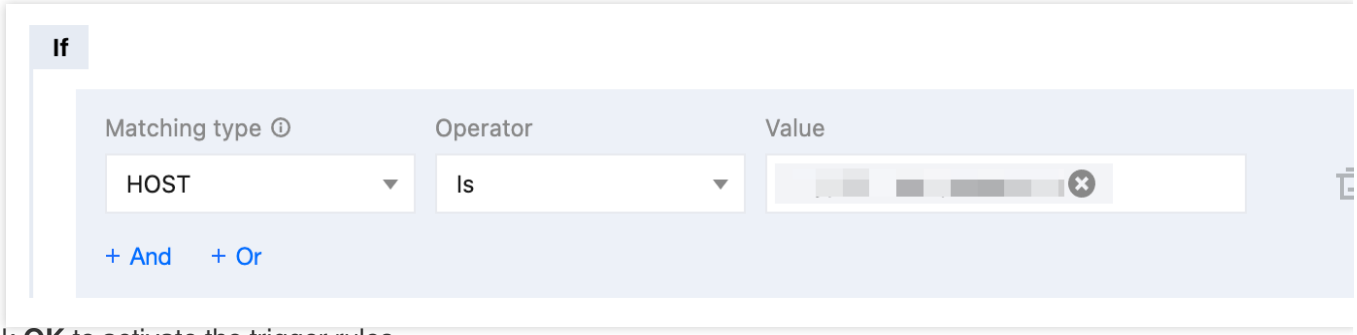
```
async function handleEvent(event) {
  const { request } = event;
  // Delete the original client region identifier from the request header to avoid
  request.headers.delete(ORIGIN_HEADER_NAME);

  let host = request.headers.get('host');

  // Attempt to obtain the country code from the geo location of the request, which
  let countryCodeAlpha2 = request.eo.geo?.countryCodeAlpha2;

  // Obtain the mapping between the region and origin server group based on the hos
  const clientOriginMap = ROUTE_CLIENT_ORIGIN_MAP[host];

  if (clientOriginMap) {
    // Attempt to obtain the origin server group name based on the country code, wh
    const originName = clientOriginMap[countryCodeAlpha2];
    if (originName) {
      // Set the request header to identify the original client region.
      request.headers.set(ORIGIN_HEADER_NAME, originName);
    } else if (clientOriginMap[OVERSEAS_AREA]) {
      // Set the request header to the identifier of regions outside the Chinese ma
      request.headers.set(ORIGIN_HEADER_NAME, clientOriginMap[OVERSEAS_AREA]);
    } else if (clientOriginMap[DEFAULT_AREA]) {
      // If no specific region is matched, use the default region, which is 'sg' in
      request.headers.set(ORIGIN_HEADER_NAME, clientOriginMap[DEFAULT_AREA]);
    }
  }

  return fetch(request);
}
```

Step 3: Configuring and Deploying Trigger Rules for the Edge Function

1. After editing the function, click **Create and deploy**. Once the function is deployed, you can directly click **Add triggering rule** to configure the trigger rules for the function.



2. In the function trigger rules, configure the trigger conditions for the function. Based on the current scenario, you can configure multiple trigger conditions using AND logic.
Here, configure the request HOST as `example.com`.
When the request URL meets the above conditions, the edge function in Step 2 will be triggered, implementing origin-pull based on client's geo location.



3. Click **OK** to activate the trigger rules.

Step 4: Configuring the Rule Engine

1. Log in to the EdgeOne console, select the site to be configured from the **Site List**, and enter the site management submenu.
2. In the left navigation bar, click **Site Acceleration** to enter the global configuration page for the site. Then, click the **Rule Engine** tab.
3. On the Rule Engine page, click **Create rule**, and then select **Add blank rule**.
4. On the rule editing page, select HOST as the matching type to match requests for a specific domain name.
Here, configure the request HOST as `example.com`.
5. On the rule editing page, click **+IF**, and based on the request header values in edge function code, configure different origin server groups.
Here, configure that when the HTTP request header `X-Forwarded-For-Origin` equals `cn`, the request will be forwarded to the origin server group in the Chinese mainland for processing; when the HTTP request header `X-Forwarded-For-Origin` equals `sg`, the request will be forwarded to the origin server group in Singapore for processing.

6. Click **Save and publish** to activate the rule engine.

Step 5: Verifying the Deployment Effect

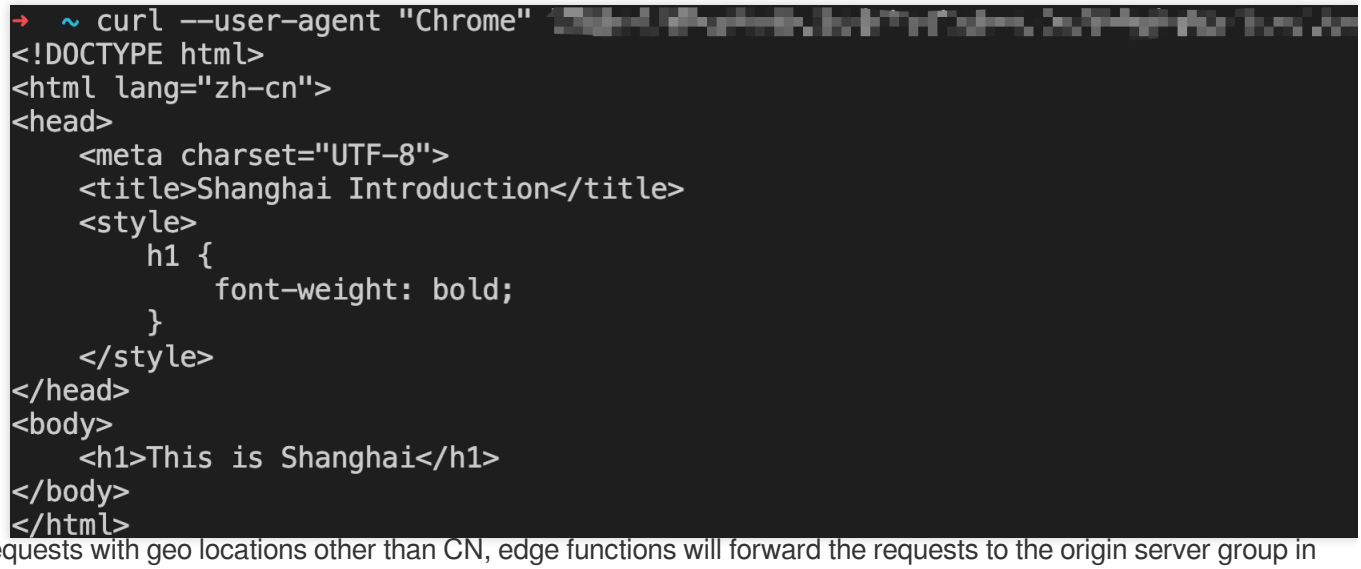
To verify the validity status of edge functions, you can use the following methods:

curl Request Test

Browser Access Test

In the Mac/Linux environment, to test the Google Chrome browser, you can run the command in the terminal: `curl --user-agent "Chrome" https://example.com`

For requests with the geo location of CN, edge functions will forward the requests to the origin server group in the Chinese mainland:



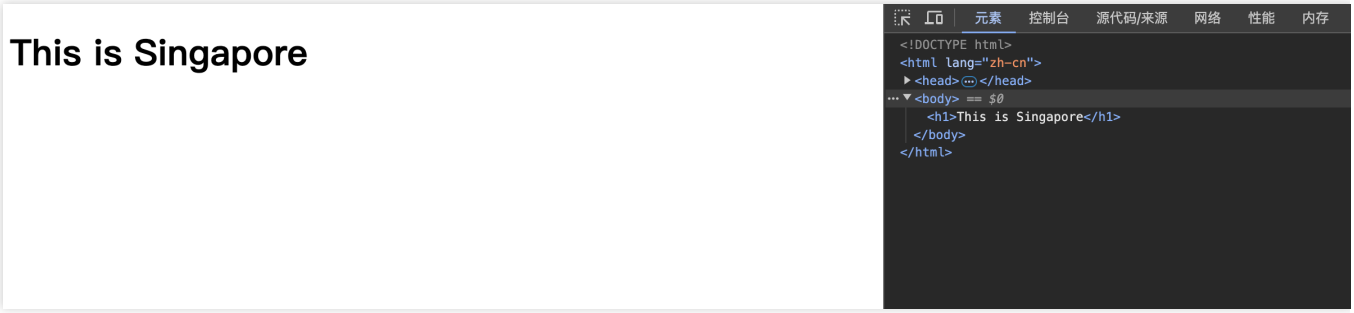
For requests with geo locations other than CN, edge functions will forward the requests to the origin server group in Singapore:

```
→  ~ curl --user-agent "Chrome" [redacted]
<!DOCTYPE html>
<html lang="zh-cn">
<head>
    <meta charset="UTF-8">
    <title>Singapore Introduction</title>
    <style>
        h1 {
            font-weight: bold;
        }
    </style>
</head>
<body>
    <h1>This is Singapore</h1>
</body>
</html>
```

Visit the test address in the Google Chrome browser: `https://example.com`

For requests with the geo location of CN, edge functions will forward the requests to the origin server group in the Chinese mainland:

For requests with geo locations other than CN, edge functions will forward the requests to the origin server group in Singapore:



More Information

EdgeOne Based on Client's Geo Location - Edge Function

EdgeOne Customization Based on Client's Geo Location - Edge Function

EdgeOne Redirection Based on the Request Region - Edge Function

APK Dynamic Packaging
EdgeOne facilitate APK dynamic packaging of Android
Feature Overview

Last updated：2023-12-05 17:35:51

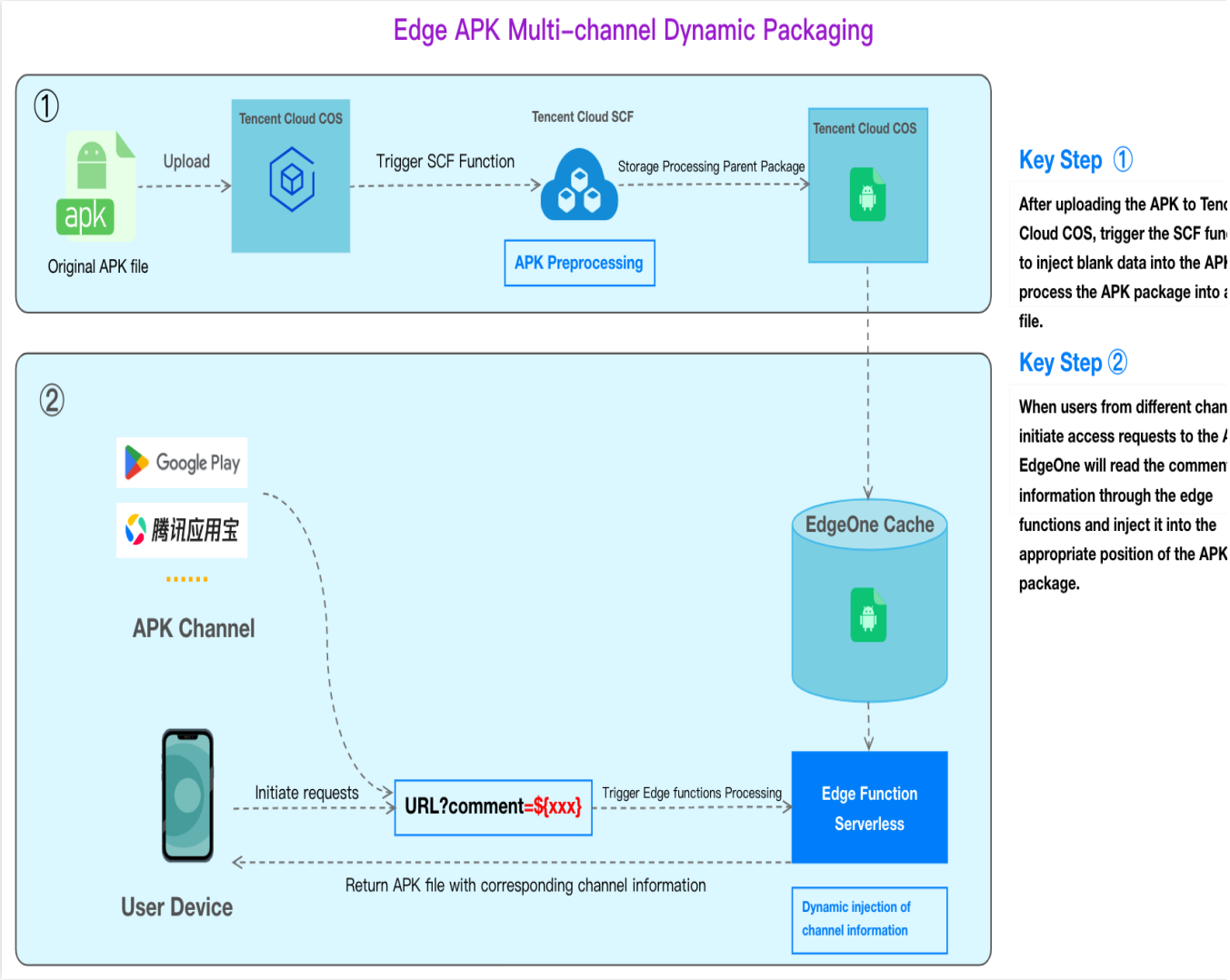
This document primarily outlines the approach to implement a dynamic packaging solution for Android APK multichannel at the edge using Tencent Cloud's EdgeOne, COS (Cloud Object Storage), and SCF (Serverless Cloud Function) products. Compared to traditional packaging methods, this solution provides a one-stop dynamic packaging and acceleration capability, reducing the maintenance complexity of multichannel APK packages and lowering the integration cost.

Background Introduction

APK (Android Application Package) is the installation package for Android applications. When an app releases a new version, it typically requires the creation of distinct channel installation packages for each distribution channel. These packages are then uploaded to the respective application markets. After users download and install the app from a specific channel, they subsequently report data. Management personnel utilize channel identifiers to track key data for each channel, such as channel download volume, conversion rates, and other critical metrics. However, the following challenges are encountered:
1. High Maintenance Cost of Channel Packages: After completing Android app development, it is typically promoted across various channels online and offline, including online app markets, affiliate networks, search engines, and offline promotions. The total number of online and offline channel partners can reach up to thousands. Maintaining a set of channel packages for each channel incurs high costs and is inefficient.
2. Difficulty in Channel Statistics: In the scenario of having multiple channels, it is necessary to calculate the installation-to-payment conversion rates for different channels. However, traditional channel analytics rely on methods like invitation codes or manual processes, leading to suboptimal results in automated statistics.
3. Inefficient Acceleration: When using CDN for APK download acceleration, each APK channel package requires individual caching, leading to uneven acceleration effects.
Therefore, against this backdrop, EdgeOne has introduced the dynamic packaging solution for multi-channel Android APKs at the edge.

Principle Introduction

The implementation of dynamic packaging for Android APK multichannel involves the following key conditions:
1. Preprocessing of APK Package: Inject blank data into the APK parent package and process it into a valid file.
2. Channel Information Injection during APK Package Download: Dynamically inject channel information into the appropriate location of the APK package when the user initiates a download operation, returning the modified APK for user download.
By employing the above approach, the decoupling of preprocessed APK packages and the channel information injection operation is achieved. The entire solution process is illustrated in the diagram below:



Solution Advantages

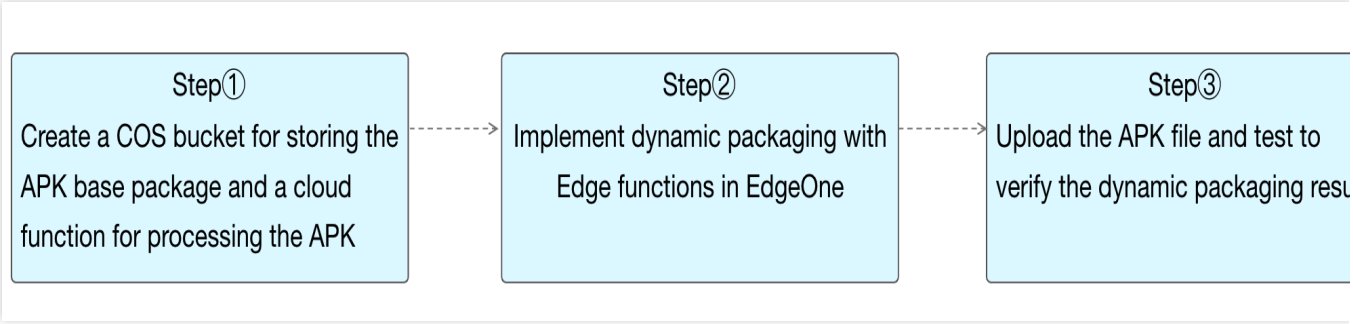
1. Reduced Channel Package Maintenance Costs: Developers only need to maintain an original Android APK parent package, eliminating the need to manage individual packages for each channel partner. EdgeOne provides default

packaging tools, allowing users to deploy easily through simple UI configuration templates, significantly enhancing the efficiency of multichannel packaging.
2. Accurate and Efficient Channel Statistics: Users can trigger edge functions by accessing a **URL with channel parameters**, dynamically inserting channel identifiers into the APK package, and returning it for user download. Simultaneously, data reporting and statistics are efficiently completed.

	EdgeOne Edge APK Dynamic Packing	VasDolly	Walle	ApkTool	Android Gra Plugin
Packing speed	Fast	Fast	Faster	Need decompression and signature,slower	Need rebuild,s
Channel information injection form	Dynamic	Static	Static	Static	Static
Channel information injection side	Edge	Origin	Origin	Origin	Origin
APK output quantity	One	Multiple	Multiple	Multiple	Multiple
packing & accleration	EdgeOne one-stop packing & acceleration	Not support acceleration	Not support acceleration	Not support acceleration	Not suppor acceleratio

Directions

Suppose you are a game manufacturer with a new Android app game (example: `v2_src.apk`) that you want to release across various channels to increase exposure and attract more players. These channels may include major app markets, app stores, social media platforms, game forums, advertising platforms, etc. Your goal is to efficiently inject channel identifiers, track channel revenue, and accelerate the download of the APK for each channel. The distribution will be centralized using the domain `apk.example.com`.



Step 1: Preprocess the Android APK Parent Package

Step 2: Write the Channel Information into the APK Package with EdgeOne Edge Functions

Step 3: Implement Test and Verify the Outcome Effectiveness

Step 1: Preprocess the Android APK Parent Package

Last updated：2023-12-05 17:48:04

This document will guide you on how to preprocess Android APK parent packages through Tencent Cloud Object Storage (COS) and Serverless Cloud Function (SCF).

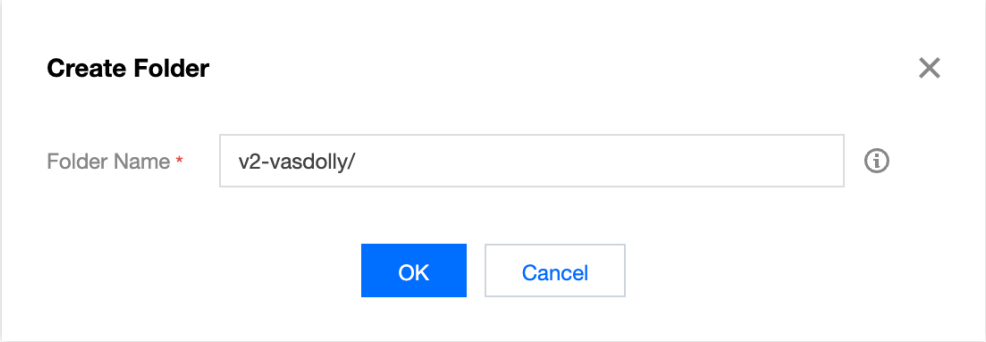
Preparation

1. Ensure that COS and SCF services are activated, and record the bucket name and region information.
2. Follow the guide on Quick Start to add a site and purchase an EdgeOne package.
3. The Domain Name for Acceleration `www.example.com` has been added in the EdgeOne console, with the origin server configured as Tencent Cloud COS.

Step 1: Upload Android APK Parent Package

In Cloud Object Storage (COS), upload the Android APK parent package.
1. Log in to the COS console. In the left menu, click on **Bucket List**.
2. On the bucket list page, click on the **Bucket Name** used to **store the APK parent package**.
3. In the file list, click on **Create Folder** to designate the directory for uploading the APK parent package, enter the folder name (example: `v2-vasdolly/`), and click **OK**.

Note:
Do not directly use the root directory as the upload directory for the APK parent package.

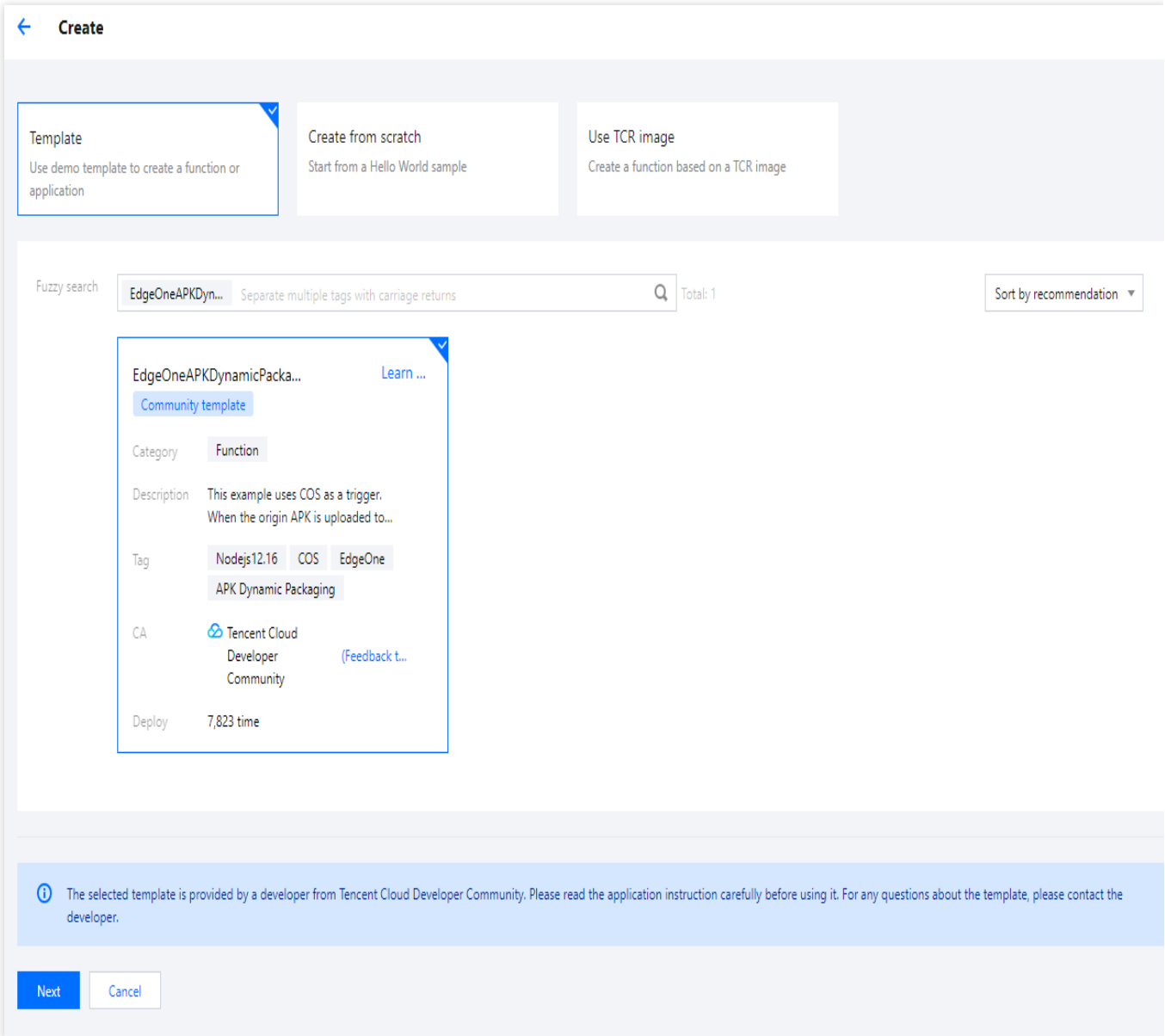


Step 2: Create a New Template Function

Create a new function in SCF via EdgeOne APK dynamic packaging template.

1. Sign in to the Serverless Cloud Function Console. In the left menu, click on **Function Service**.

2. On the function service page, click on **Create** and choose **Template**. In the fuzzy search, enter "EdgeOne APK Dynamic Packaging", select it, and click **Next**.



3. On the **Function Configuration** page, configure the following parameters:

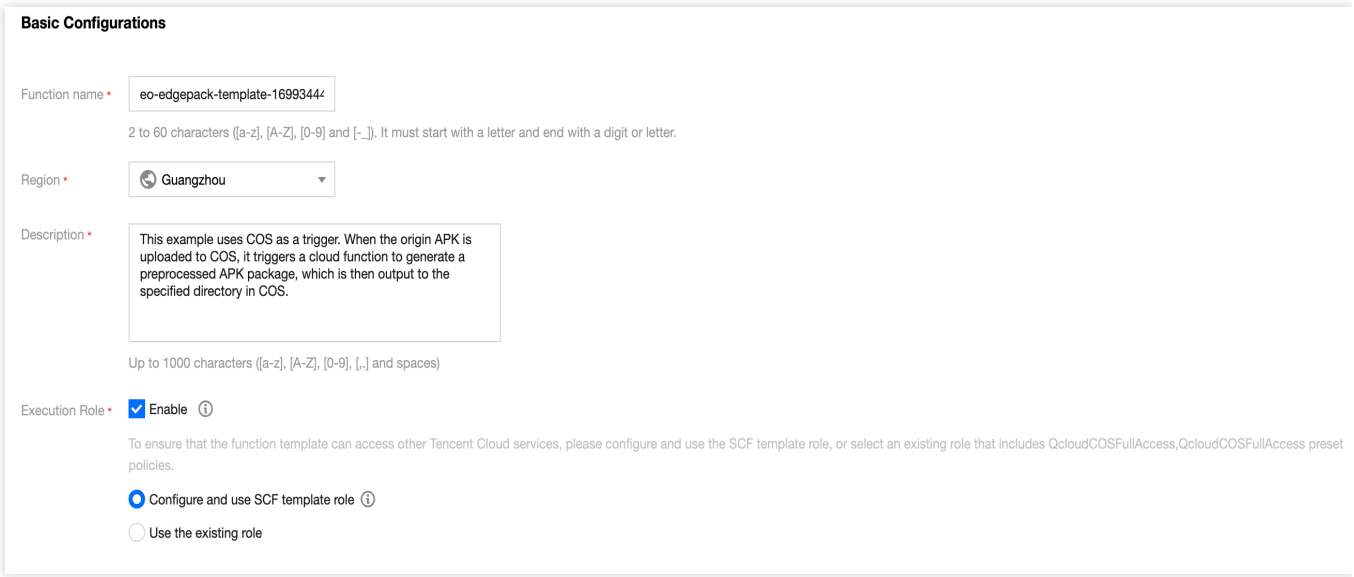
Basic Configuration

Function name: When creating the function, a name will be generated automatically. You can choose to customize it for easy recognition.

Region: Choose the region where the COS bucket from Step 1 is located, example: Guangzhou.

Description: Describe the purpose of this function.

Execution Role: By default, **Enable, Configure and use SCF template role** is selected. If an existing role is used, ensure it includes the preset policies `QcloudCOSFullAccess` .



Function Code: The template has built-in default function code for processing Android APK parent packages, and no modification is required.

Environment Configuration:

Click on **Advanced Configuration**, select **Environment Configuration**, and add the following keys and corresponding values to the environment variables. Keep the rest of the configurations as default:

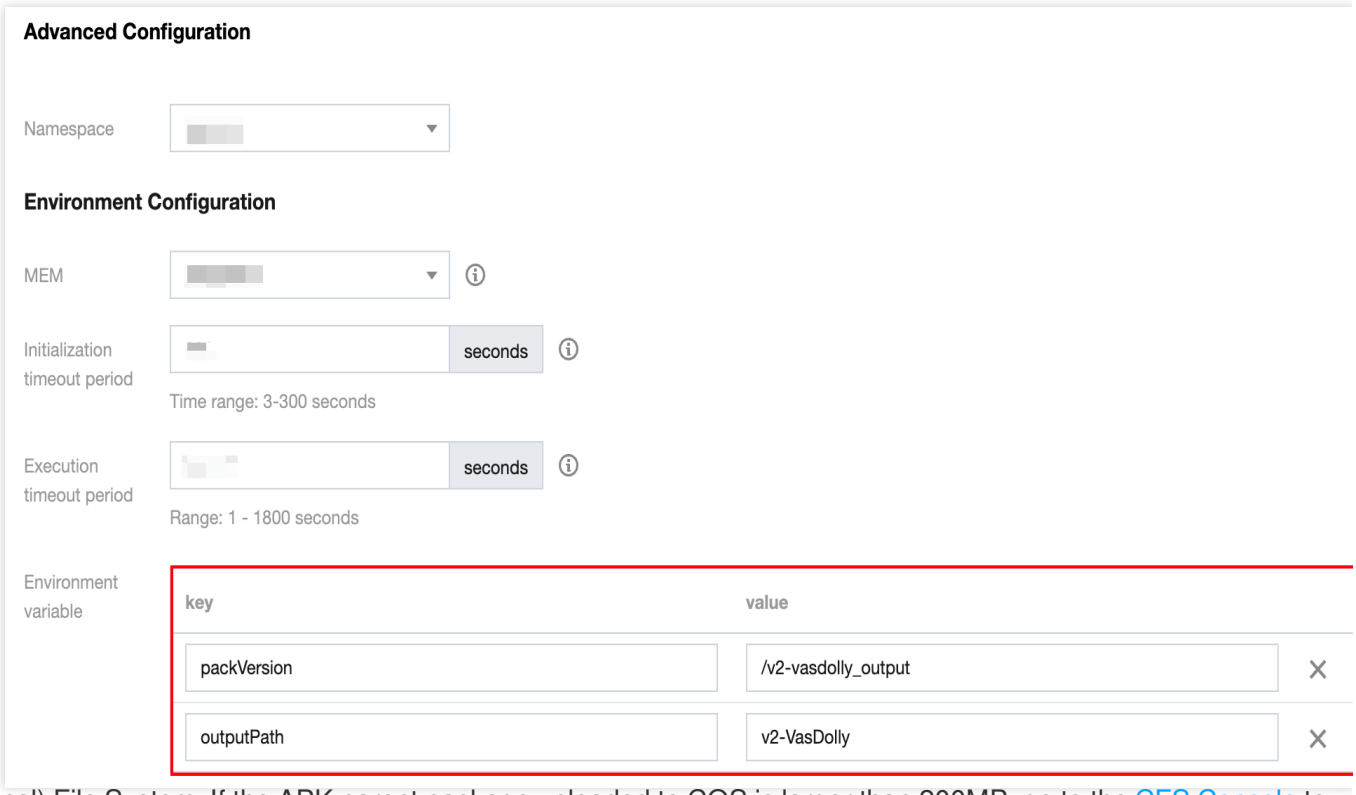
outputPath (Required): Customize the directory in the COS bucket where the Cloud Function SCF outputs the processed APK parent package, for example, `/v2-vasdolly_output`.

packVersion (Required): Information about the signature version used for different APK versions. Enter the following values for different signature versions:

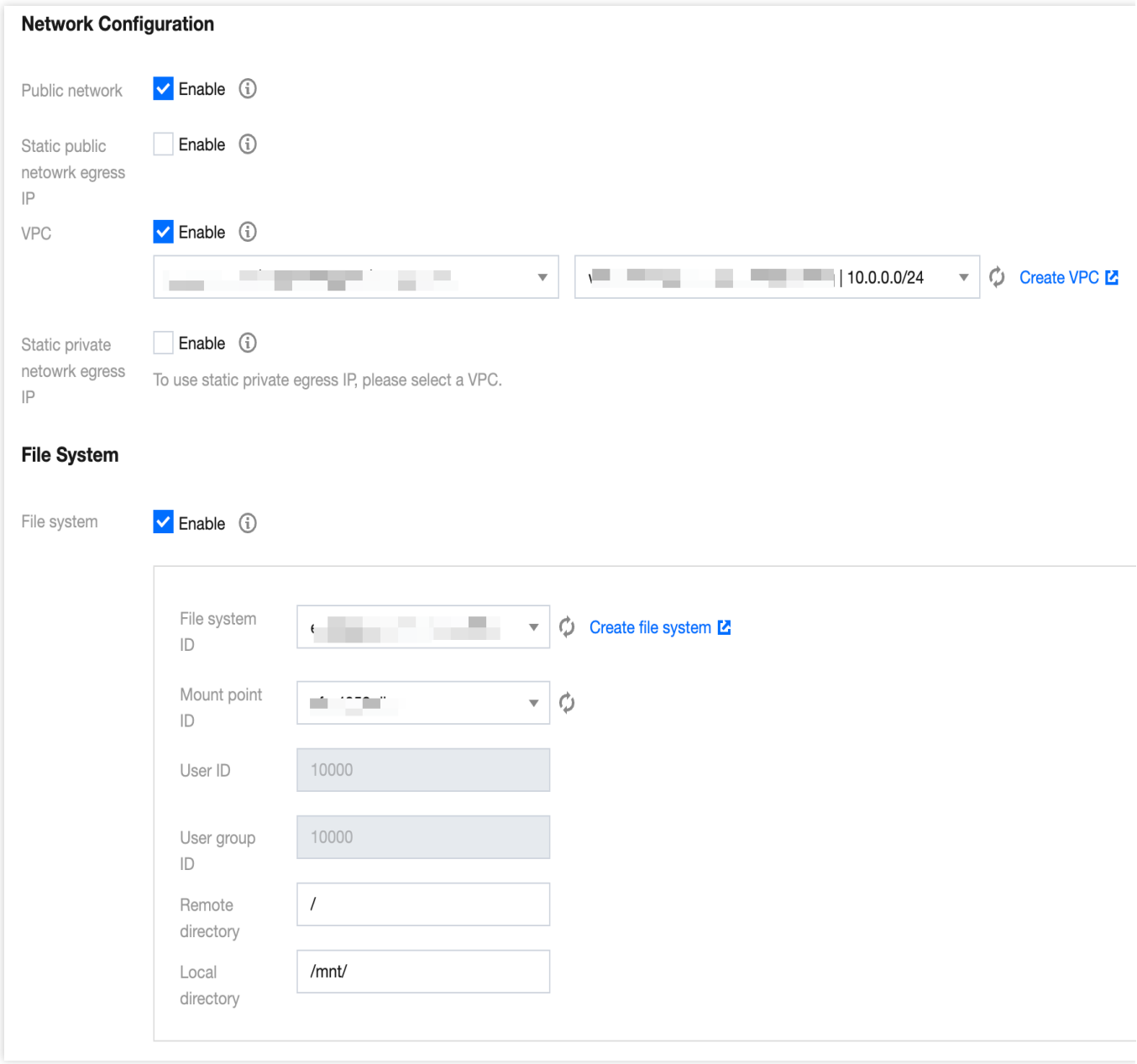
APK Signature Version	packVersion Value
v1	v1
v2	Please enter v2-VasDolly, v2-Walle, or v2-Custom: v2-VasDolly: Store the channel information in the ID-Value pair with the ID `0x881155ff` (VasDolly default). v2-Walle: Store the channel information in the ID-Value pair with the ID `0x71777777` (Walle default). v2-Custom: Store the channel information in the ID-Value pair with the ID specified by the `blockId` environment variable. v2-Custom: The channel information is stored in the ID-Value pair with ID `blockId` (specified by the blockId environment variable).

blockId(Optional): If using the v2-Custom method for preprocessing, specify the blockId.

Examples:



(Optional) File System: If the APK parent package uploaded to COS is larger than 200MB, go to the CFS Console to enable the CFS service and file system for expanding the local storage space of SCF.



Note:

Due to the limitations on the SCF side, each cloud function has a temporary disk space of 500MB during execution. When processing APK files, both the original APK file and the processed APK file coexist in the disk. Therefore, for processing excessively large APK files, it is necessary to mount an additional file storage system. For details, see Mounting CFS File System.

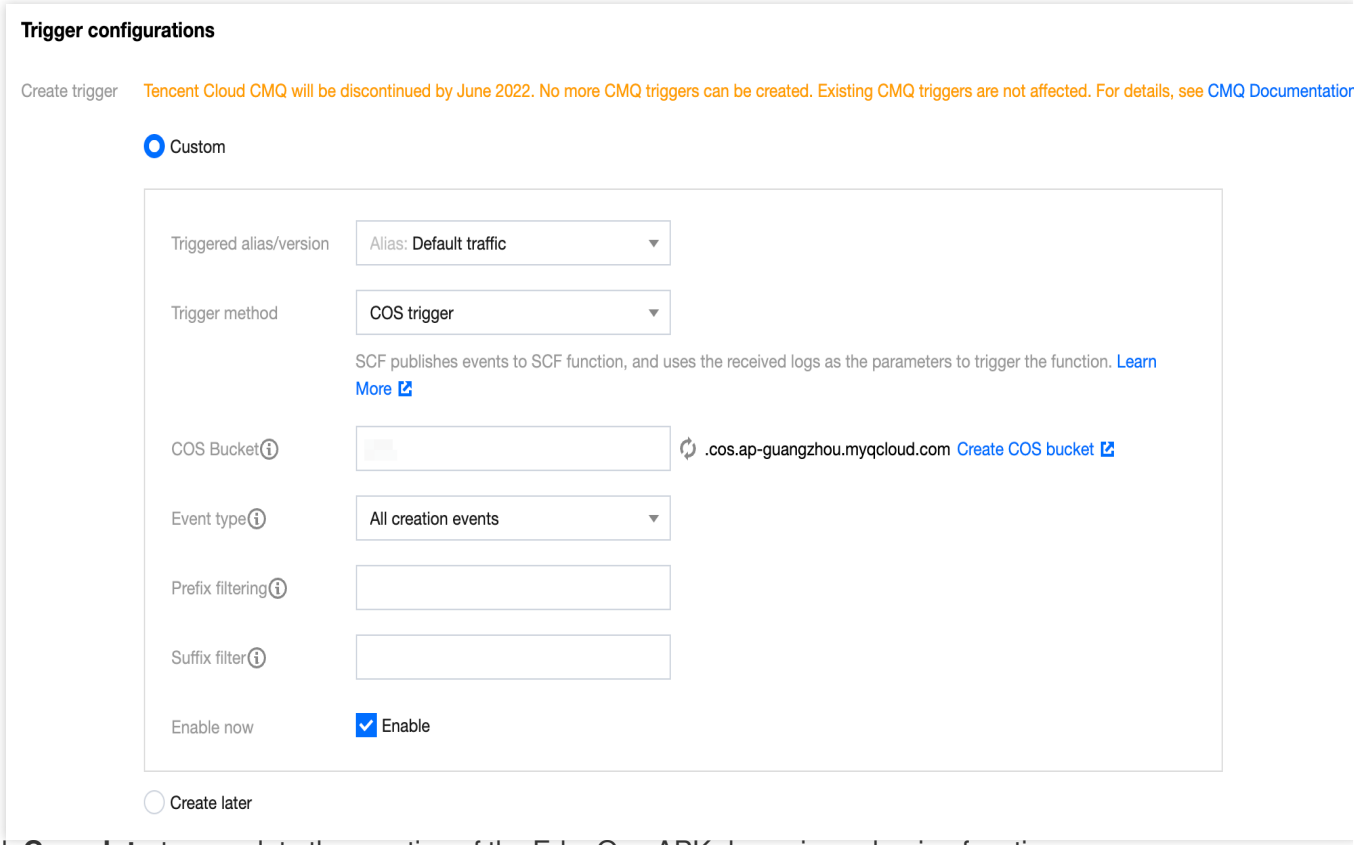
Trigger Management

In the trigger configuration, select the bucket for the COS bucket in the same region as that of the SCF. Enter the bucket name for fuzzy search, for example: `apk-test-1251557890.cos.ap-guangzhou.myqcloud.com`. Keep the other configurations as default.

Trigger Mode: Choose COS trigger.

COS Bucket: Select the COS bucket where the parent package resides in this available zone.

Event Type: Choose All Created Events.
Prefix Filter: Please enter the directory where the APK parent package is uploaded. For example, if your parent package is in the `v2-vasdolly` directory, enter `v2-vasdolly/` .
Suffix Filter: Please enter `.apk` .
Once the above information is filled out, the SCF function will only be triggered when files with a `.apk` suffix are uploaded to the specified `v2-vasdolly/` directory in the designated COS bucket.



4. Click **Complete** to complete the creation of the EdgeOne APK dynamic packaging function.

Note:

Please proceed to Step 2: Write the Channel Information into the APK Package with EdgeOne Edge Functions.

Step 2: Write the Channel Information into the APK Package with EdgeOne Edge Functions

Last updated：2025-04-25 10:50:08

Through EdgeOne edge function, we can dynamically write channel information into the APK package. Users only need to access the domain bound to the edge function and trigger the appropriate configuration to enable the edge function, achieving dynamic packaging and accelerated distribution of the APK.

Step 1: Add an Acceleration Domain Name for Enhanced Distribution Speeds

Please follow the instructions in Adding A Domain Name for Acceleration to add an acceleration domain, for example: `www.example.com` , and configure the origin server to the COS where the Android APK parent package is located, as shown below:

Note:

This domain will be used to access and download the APK installation package.

Step 2: Create an Edge Function for Triggering Channel Information Writing

1. Follow the instructions in Function Management to create an edge function and copy the following code into the function code.

```
const CUSTOM_BLOCK_VALUE_LENGTH = 10240;
const APK_SIGNING_BLOCK_MAGIC_LENGTH = 16;
const APK_SIGNING_BLOCK_OFFSET_LENGTH = 8;

const APK_COMMENT_LENGTH = 512;

class EdgePack {
  totalSize;
  signVersion;
  centralDirectoryOffset;
  customBlockValueStart;
  customBlockValueEnd;
  rangeRelativeOffset;
```

```
  customInfo;

  constructor() {
    this.totalSize = null;
    this.signVersion = null;
    this.centralDirectoryOffset = null;
    this.customBlockValueStart = null;
    this.customBlockValueEnd = null;
    this.rangeRelativeOffset = null;
    this.customInfo = null;
  }

  async handle(event) {
    const { request } = event;

    const headers = new Headers(request.headers);

    const modifiedRequest = new Request(request, { headers });

    if (!this.checkRequest(modifiedRequest)) {
      return;
    }

    let response = null;
    try {
      const headRequest = new Request(modifiedRequest.url, {
        method: 'HEAD',
        headers: modifiedRequest.headers,
      });
      response = await fetch(headRequest);
    } catch (err) {
      const error = {
        code: 'FETCH_ORIGIN_ERROR',
        message: err?.message,
      };
      response = new Response(JSON.stringify(error), {
        status: 590,
      });
    }

    if (!this.checkResponse(response)) {
      return event.respondWith(response);
    }

    response.headers.set('Cache-Control', 'max-age=0');

    const streamResponse = new Response(
```

```
      await this.combineStreams(modifiedRequest),
      response
    );

    event.respondWith(streamResponse);
  }

  getRelativeOffset(response) {
    const start = this.customBlockValueStart;
    const end = this.customBlockValueEnd;

    const range = response.headers.get('Content-Range');

    if (!range) return start;

    const match = range.match(/bytes\\s*(\\d*)-(\\d*)/i);
    if (!match || match?.length < 2) {
      return start;
    }

    if (+match[2] < start || +match[1] > end) {
      return null;
    }

    return start - +match[1];
  }

  checkRequest(request) {
    if (request.method !== 'GET') {
      return false;
    }

    if (request.headers.has('Range')) {
      return false;
    }

    const { pathname, searchParams } = new URL(request.url);

    const comment = searchParams?.get('comment');

    if (!pathname.endsWith('.apk') || !comment) {
      return false;
    }

    this.customInfo = comment;
    return true;
  }
```

```
  checkResponse(response) {
    if (response.status !== 200 && response.status !== 206) {
      return false;
    }

    const contentLength = response.headers.get('Content-Length');

    if (response.body === null || contentLength === null) {
      return false;
    }

    this.totalSize = Number(contentLength);

    const cosOffsetHeader = response.headers.get('x-cos-meta-edgepack-offset');
    const cosTypeHeader = response.headers.get('x-cos-meta-edgepack-type');

    if (!cosOffsetHeader || !cosTypeHeader) {
      return false;
    }

    this.signVersion = cosTypeHeader;
    this.centralDirectoryOffset = Number(cosOffsetHeader);

    if (this.signVersion === 'v1') {
      this.customBlockValueStart = this.totalSize - APK_COMMENT_LENGTH;
      this.customBlockValueEnd = this.totalSize;
    } else {
      this.customBlockValueStart =
        this.centralDirectoryOffset -
        CUSTOM_BLOCK_VALUE_LENGTH -
        APK_SIGNING_BLOCK_MAGIC_LENGTH -
        APK_SIGNING_BLOCK_OFFSET_LENGTH;
      this.customBlockValueEnd = this.centralDirectoryOffset;
    }

    this.rangeRelativeOffset = this.getRelativeOffset(response);

    if (this.rangeRelativeOffset === null) {
      return false;
    }

    return true;
  }

  async combineStreams(request) {
    const { readable, writable } = new TransformStream();
```

```
    this.handleStream(request, writable);
    return readable;
  }

  async handleStream(request, writable) {
    const comment = this.customInfo;
    const relativeOffset = this.rangeRelativeOffset;

    const encoder = new TextEncoder();
    const section = encoder.encode(comment);

    try {
      const apkHeader = await this.apkHeaderStream(request);

      try {
        await apkHeader.pipeTo(writable, {
          preventClose: true,
        });
      } catch (e) {
        console.error('HEADER_STREAM_ERROR: ', e);
      }

      // Return to Blob data
      const apkBody = await this.apkBodyStream(
        request,
        section,
        relativeOffset
      );

      const apkBodyStream = apkBody.stream();

      try {
        await apkBodyStream.pipeTo(writable, {
          preventClose: true,
        });
      } catch (e) {
        console.error('BODY_STREAM_ERROR: ', e);
      }

      const apkTail = await this.apkTailStream(request);

      try {
        await apkTail.pipeTo(writable, {
          preventClose: true,
        });
      } catch (e) {
        console.error('TAIL_STREAM_ERROR: ', e);
```

```
      }
    } catch (err) {
      console.error('HANDLE_STREAM_ERROR: ', err);
    } finally {
      let writer = writable.getWriter();
      writer.close();
      writer.releaseLock();
    }
  }

  async apkHeaderStream(request) {
    const headers = new Headers(request.headers);
    headers.set('Range', `bytes=0-${this.customBlockValueStart - 1}`);

    //Obtaining the part before the signature block.
    const headResponse = await fetch(request, {
      headers: headers,
    });

    return headResponse.body;
  }

  async apkBodyStream(request, section = null, relativeOffset = 0) {
    const headers = new Headers(request.headers);
    headers.set(
      'Range',
      `bytes=${this.customBlockValueStart}-${this.customBlockValueEnd - 1}`
    );

    const middleResponse = await fetch(request, {
      headers: headers,
    });

    const reader = middleResponse.body.getReader();

    let outputBuffers = [];
    try {
      let handledBytes = this.customBlockValueStart;
      while (true) {
        const result = await reader.read();

        if (result.done) {
          console.log('APK_BODY_STREAM_DONE');
          break;
        }

        const startByteOffset = handledBytes;
```

```
        const buffer = result.value;
        handledBytes += buffer.byteLength;

        const min = Math.max(startByteOffset, relativeOffset);
        const max = Math.min(relativeOffset + section.byteLength, handledBytes);

        if (min < max) {
          const bufferStart = min - startByteOffset;
          const sectionStart = min - relativeOffset;
          const sectionEnd = max - relativeOffset;

          const replacement = section.subarray(sectionStart, sectionEnd);

          new Uint8Array(buffer).set(replacement, bufferStart);
        }

        outputBuffers.push(buffer);
      }
    } catch (err) {
      console.error('APK_BODY_STREAM_ERROR: ', err);
    }
    return new Blob(outputBuffers);
  }

  async apkTailStream(request) {
    const headers = new Headers(request.headers);
    headers.set(
      'Range',
      `bytes=${this.customBlockValueEnd}-${this.totalSize - 1}`
    );

    const tailResponse = await fetch(request, {
      headers: headers,
    });

    return tailResponse.body;
  }
}

async function handleEvent(event) {
  const edgepack = new EdgePack();
  await edgepack.handle(event);
}

addEventListener('fetch', handleEvent);
```

2. After deploying the function, configure the trigger rule under Function Management as directed, where the HOST value is the acceleration domain name created in Step 1, as shown below:

3. Click **OK** to complete the creation of the trigger rule. When users access the domain `www.example.com` with a file suffix of `.apk`, it will trigger the edge function for dynamic packaging.

Note:

Please proceed to Step 3:Implement Test and Verify the Outcome Effectiveness.

Step 3: Implement Test and Verify the Outcome Effectiveness

Last updated：2025-04-25 10:46:18

Step 1: Verify SCF's Preprocessing of Android APK Parent Package

1. Log in to the COS console. In the left menu, click on **Bucket List**.
2. On the bucket list page, click on the **Bucket Name** used to **store the APK parent package**.
3. In the file list page, click on the `v2-vasdolly/` directory, click **Upload Files** and select a file ending with `.apk` , for example**:** `v2_src.apk` . Click **Upload**.

4. If the SCF has successfully processed the Android APK parent package, a new output directory will be generated at the same level as the COS upload directory. The specific path is the directory filled in the `outputPath` in the Create Template Function, for example, `/v2-vasdolly_output` . Click on the **directory name** to enter it, and you will see the SCF has preprocessed the new APK parent package.

Step 2: Verify the Channel Information Written into the Android APK Package through EdgeOne Edge Functions

Enter a URL with channel information in the browser, for example, `http://www.example.com/v2_src.apk?comment=test` . This will trigger the edge function to dynamically inject the channel information into the specified location. In this case, "comment" is the channel parameter defined in the Creation of the Edge Function for Injecting Channel Information. Using the v2-VasDolly method as an example, you can use the VasDolly tool to read the dynamically injected channel information.

Data Analysis and Alerting
Configuring EdgeOne Security Event Alarms via TCOP

Last updated：2024-09-24 18:09:35

Background

EdgeOne, in collaboration with Tencent Cloud Observability Platform (TCOP), offers flexible alarm solutions for security events such as denial-of-service (DDoS) attacks, challenge collapsar (CC) attacks, and DDoS attack blocking. Users can leverage TCOP's alarm capabilities to set detailed alarm trigger rules and receive alarms through various notification channels, including telephone, SMS, email, WeChat, and VIP customer support groups. This significantly improves response speed and handling efficiency for security threats.

Note:

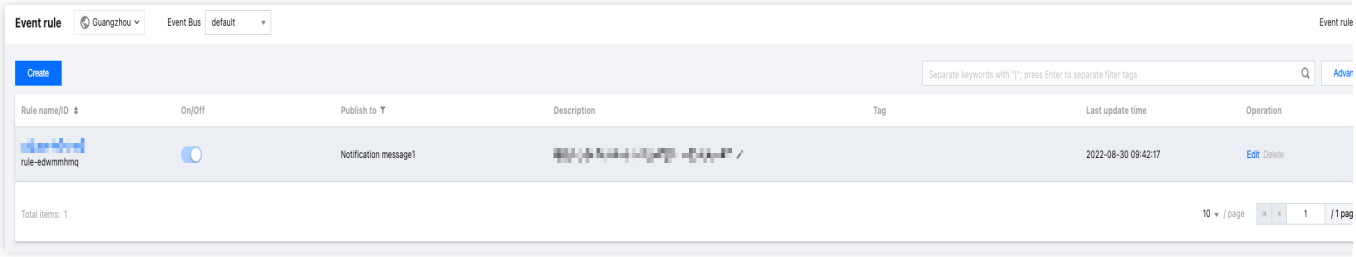
When you select telephone and SMS as alarming channels on TCOP, related fees may be incurred, which are charged by TCOP.

Applicable Scenario

This document is applicable to all users who have integrated EdgeOne and need to configure security event alarms.

Default Alarm Policy

Once you have connected a domain name/L4 proxy instance to EdgeOne, TCOP will, by default, push alarm messages to the email and SMS configured for your Tencent Cloud root account when a security event occurs. You can view the rules for default alarms of cloud service events in TCOP - Event Bus - Event Rules.



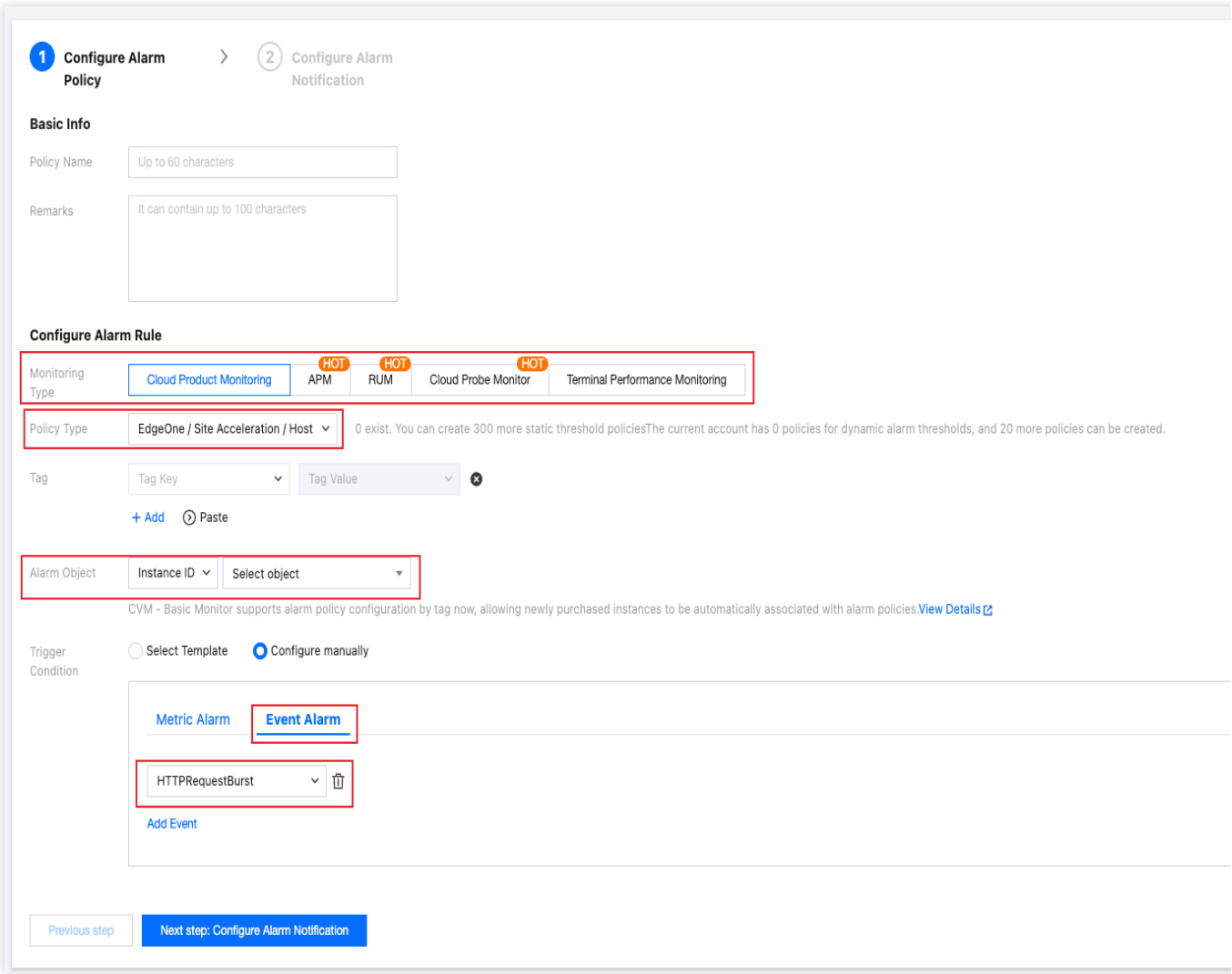
Operation Step

Step 1: Configure an Alarm

1. Log in to the TCOP console, in the left navigation bar, choose **Alarm Management** > **Alarm Configuration**, and click **Create Policy**.
2. The detailed configuration of the alarm policy is as follows:
2.1 Select **Cloud Product Monitoring** for the monitoring type.
2.2 Select **EdgeOne** / **Site Acceleration** / **Host** for the policy type. Different security event alarms require different policy types. See the table below for details:

EdgeOne Security Event Type	TCOP Alarm Policy Type	Configuration Meaning
HTTPRequestBurst	EdgeOne / Site Acceleration / Host	Alarm for CC attacks on the **specified domain name**
DDoS Attack / DDoS Attack Blocked	EdgeOne / L4 Proxy / Instance	Alarm for DDoS attacks/blocking events on the **specified L4 proxy instance**
	EdgeOne / Plan	Alarm for DDoS attacks/blocking events on the EdgeOne plan of the **specified L7 business**

2.3 Select the domain name list you want to monitor as the alarm object.
2.4 Select **Event Alarm** for the trigger condition.
2.5 Select **HTTPRequestBurst** from the drop-down list.
2.6 For other related configurations, refer to Creating Alarm Policy.
3. Click **Next step: Configure Alarm Notification**.



Step 2: Configure Alarm Notifications

1. Determine whether the **system preset notification template** meets expectations. If you need a custom notification template, refer to Creating Notification Template.
2. After selecting the required notification template, click **Complete** to save the configuration.

References

EdgeOne Security Events and Corresponding Handling Suggestions

The following is a list of security events that could be triggered by EdgeOne, including event types, event descriptions, suggestions, and more.

Event Type	Event Description	Suggestion
HTTPRequestBurst	EdgeOne has detected a sudden increase in HTTP requests to the domain name, possibly due to a CC attack.	1. Monitor your business availability. You can also check recent traffic and request details on the EdgeOne console - Metrics Analysis

	Note: The trigger condition is that the rate of HTTP requests exceeds 1,000 queries per second (QPS), and this increase is beyond the baseline of normal traffic predicted by the platform's intelligent learning algorithm.	page to determine whether the spike in traffic is part of normal business activity. 2. If you determine that the sudden increase in traffic is not part of normal business activity and the current security policy does not cover the characteristics of the attack, it is recommended to modify and tighten the Web protection policy. 3. If you determine that the sudden increase in traffic is part of normal business activity, you can ignore this alarm. Additionally, it is recommended to loosen the Web Protection - Adaptive Frequency Control Limit Level or switch to observation mode.
DDoSAttack	EdgeOne has detected that the IP address serving you is under a DDoS attack. **Note:** The trigger condition is that the detected DDoS attack bandwidth exceeds the DDoS attack traffic alarm threshold configured by a customer in the EdgeOne console (default value is 100 Mbps).	Monitor your business availability. You can also click the **L3/4 DDoS Attack Protection Bandwidth** tab on the EdgeOne console - Metrics Analysis page, and then click the **Number of DDoS Attack Events** tab at the top to view details of the corresponding attack events.
DDoSAttackBan	The IP address serving you has been blocked by the ISP due to a DDoS attack.	Contact us.