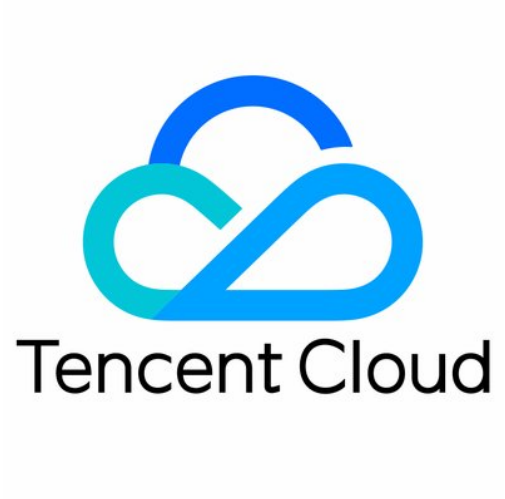
Tencent Cloud EdgeOne

Best Practices

Product Documentation

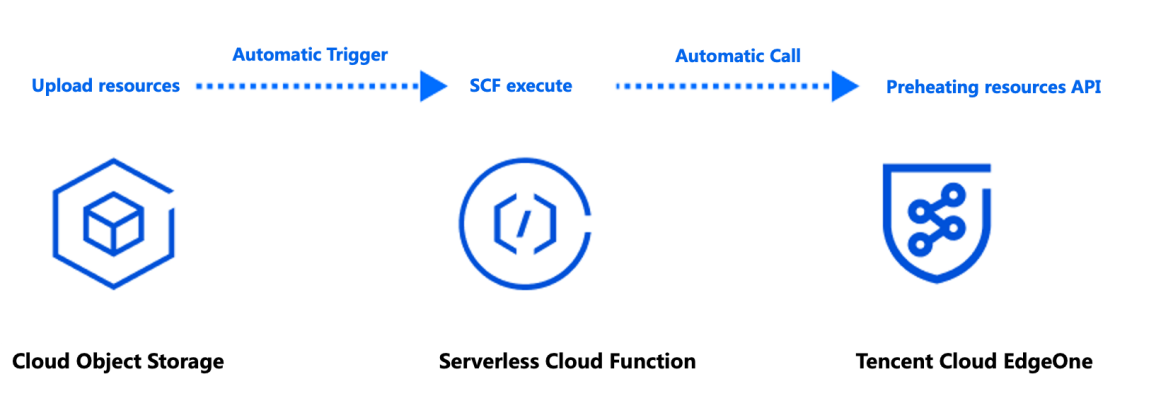


Contents

Best Practices
EdgeOne initiates Automatic Warm-up

Last updated：2023-12-06 10:43:20

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.

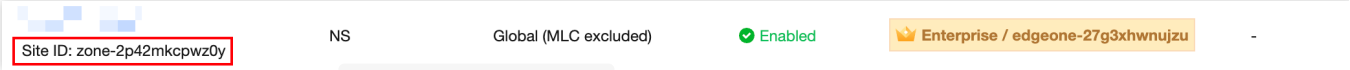
Directions

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

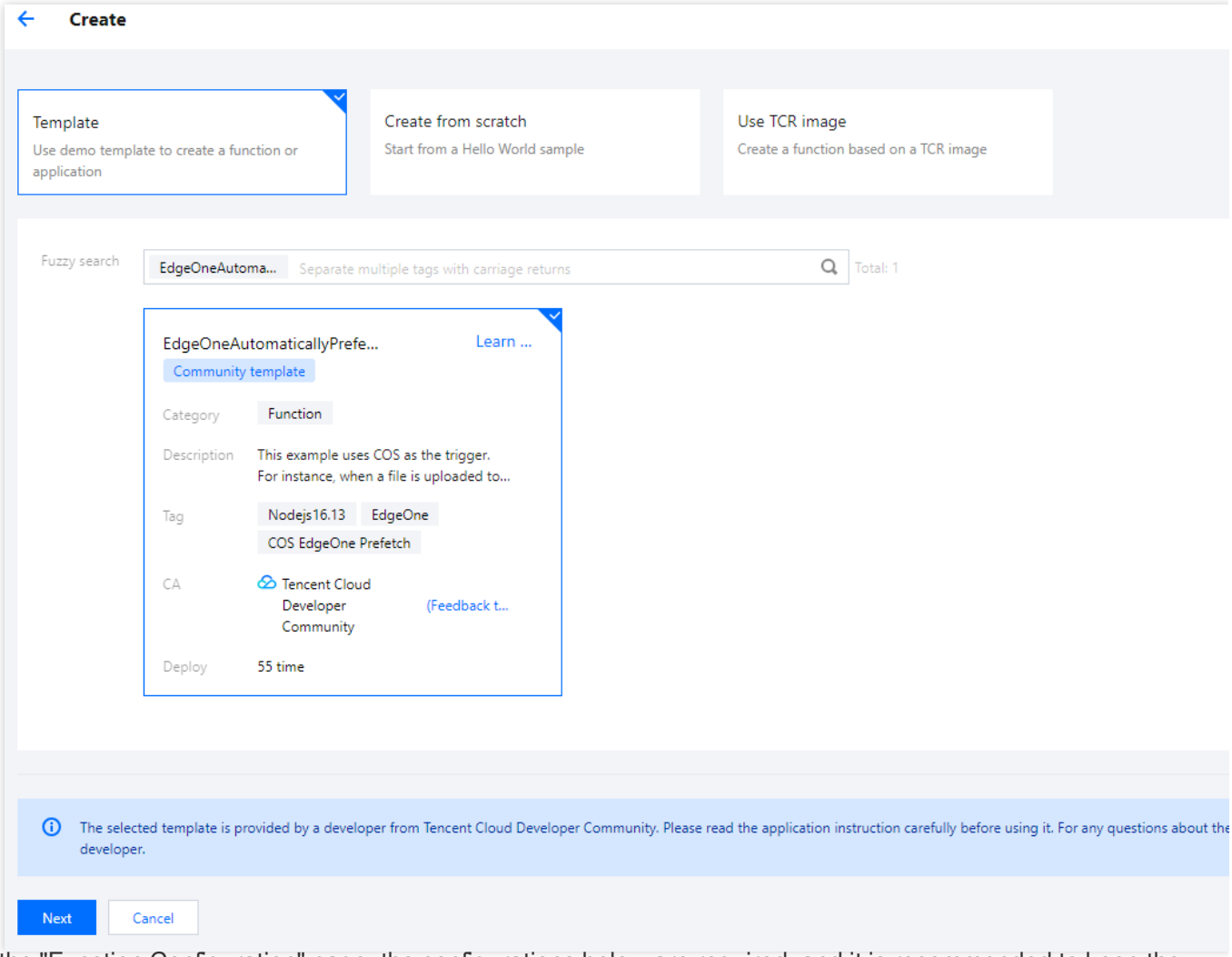
1. Ensure that COS and SCF services are activated, and record the bucket name and region information.
2. Follow the Quick Start to add your site, purchase the EdgeOne package, and obtain the site ID. The site ID can be found and copied from the site list after site access, for example, zone-2p42mkcpwz0y.



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.

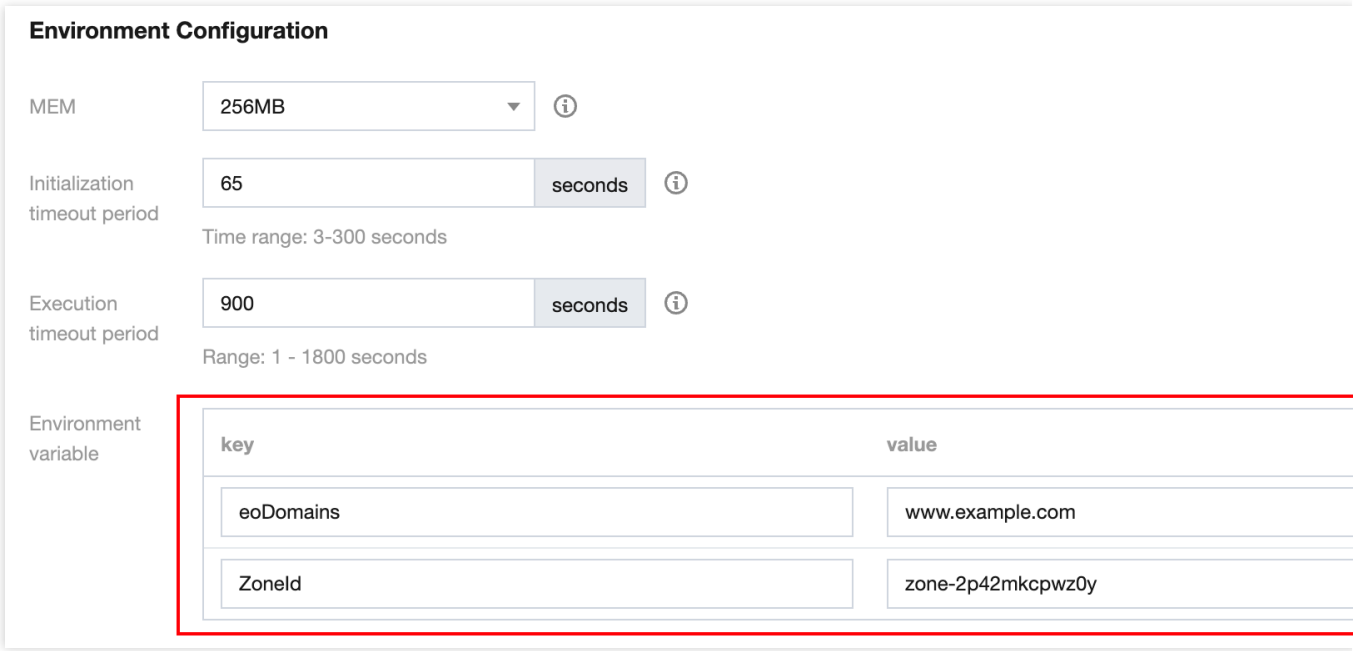
Function Codes: The template already includes default function code implementing the EdgeOne automatic pre-warming capability. No modifications are necessary.

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:

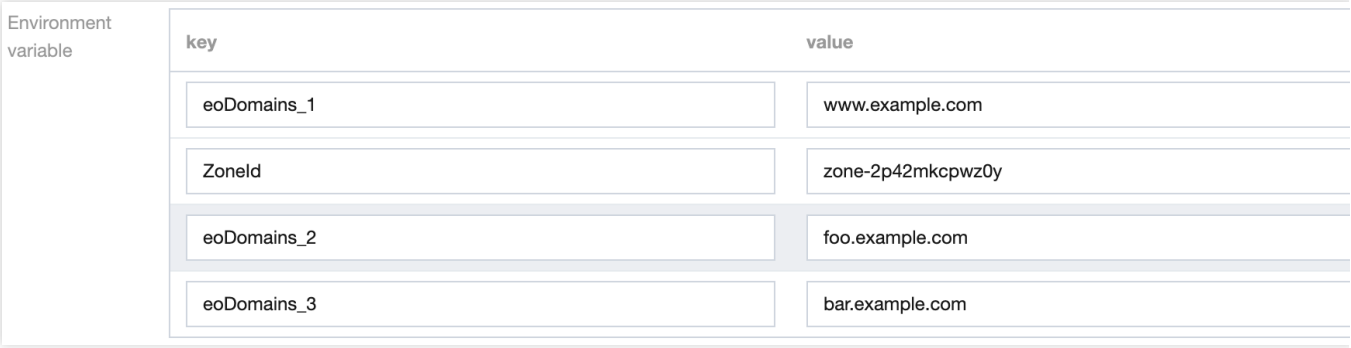
ZoneId: Fill in the ZoneId of the domain site `example.com` that needs automatic pre-warming. See the Preparation for obtaining the Site ID.

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



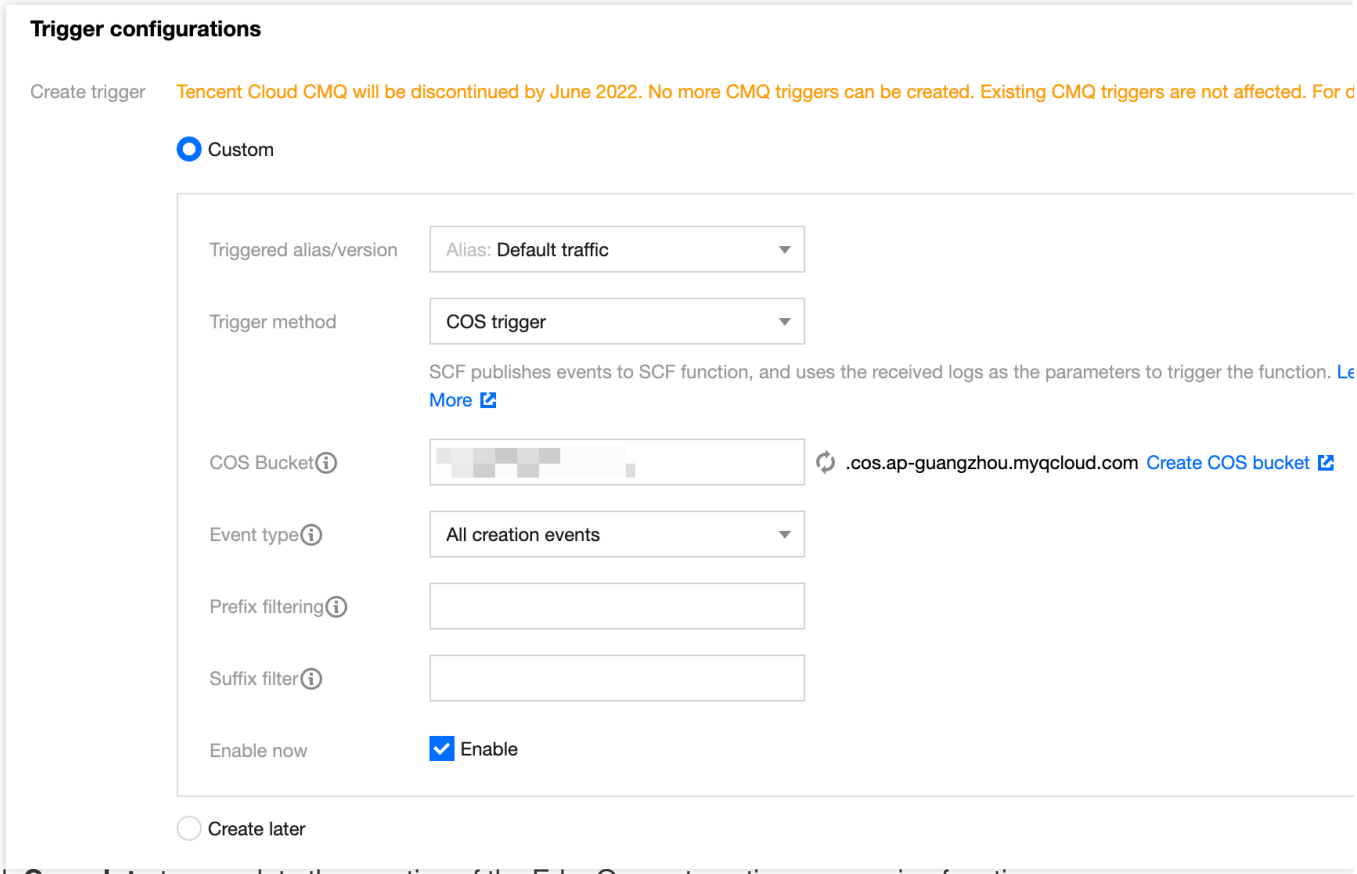
Note:

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.



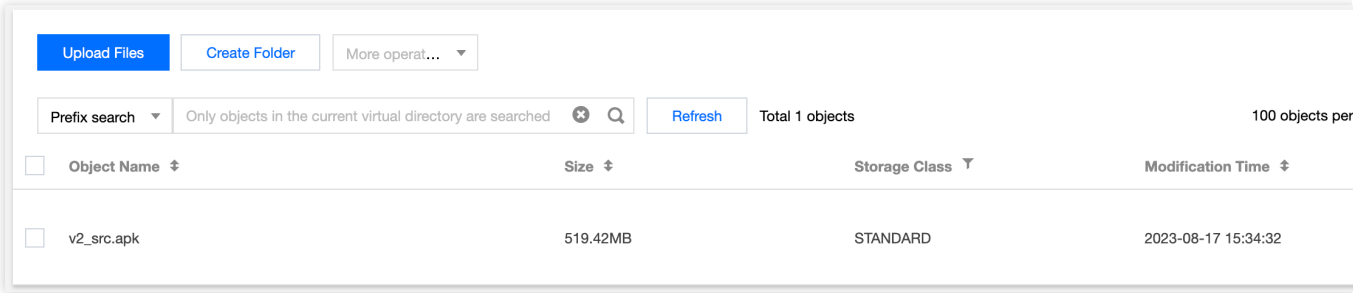
4. Click **Complete** to complete the creation of the EdgeOne automatic pre-warming function.

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**.

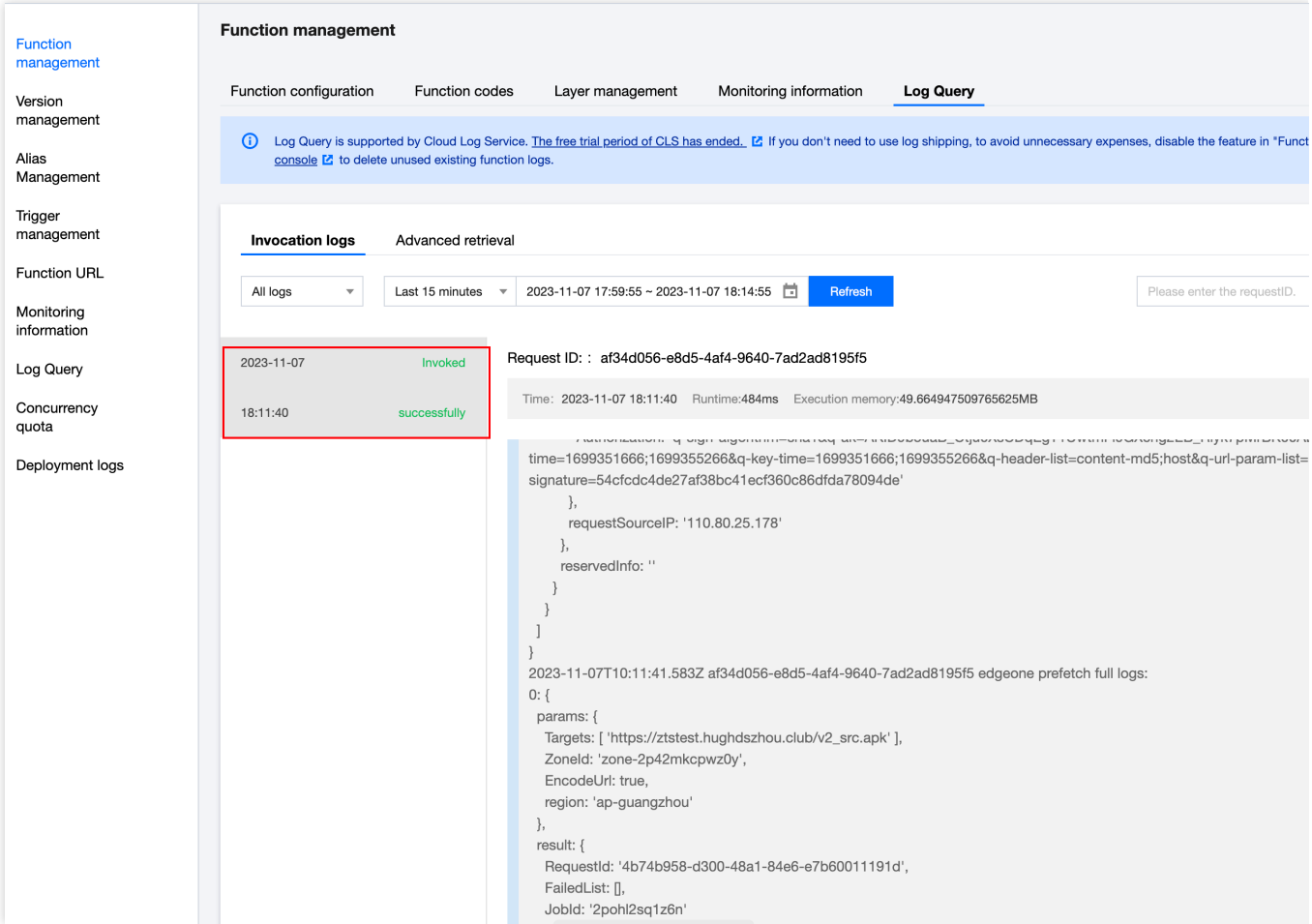
3. In the file list page, enter the root directory `prefetch-cos-1251558888.cos.ap-guangzhou.myqcloud.com` .

4. Click **Upload Files** and upload a file for the first time, for example, v2_src.apk, and then click **Upload**.



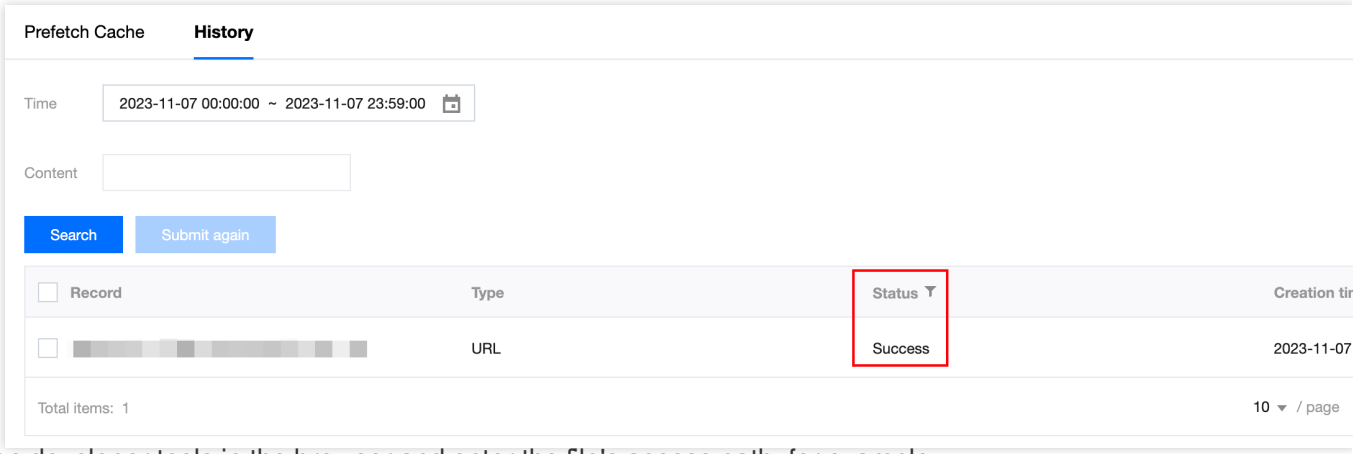
5. After successful file upload, in the SCF Console, click on the **Function Name** created in Step 1.

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.

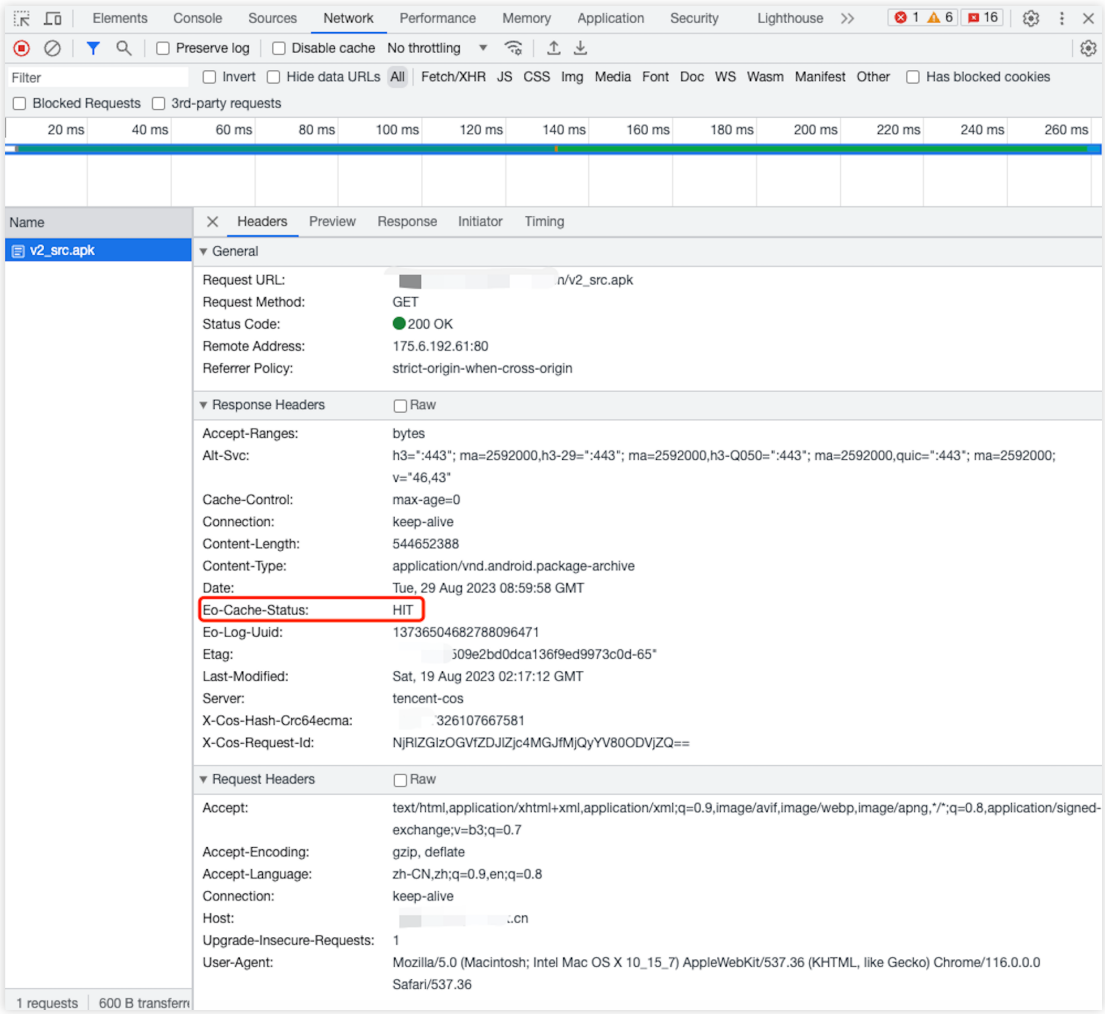


7. Go to the EdgeOne Console, enter the current site `example.com` , and click on **Site Acceleration** > **Cache Prefetching**.

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.



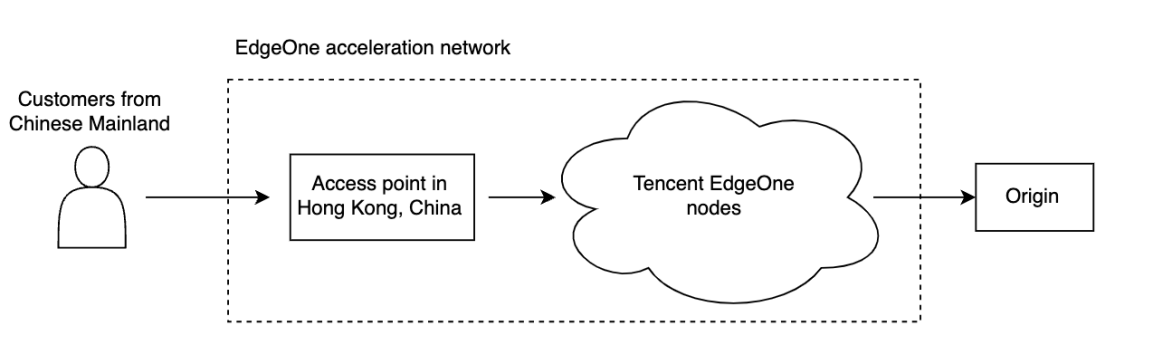
Cross-regional Secure Acceleration (Oversea Sites)

Last updated：2023-08-21 14:11:10

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

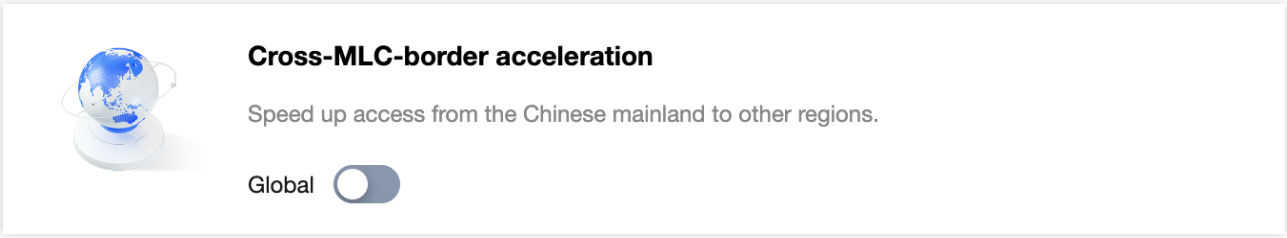
1. Follow the site access guide to add a site, purchase the EdgeOne Enterprise plan, and set the site acceleration area to Global (MLC excluded).
2. Contact the business department to enable the Cross-MLC-border acceleration function.

Note：

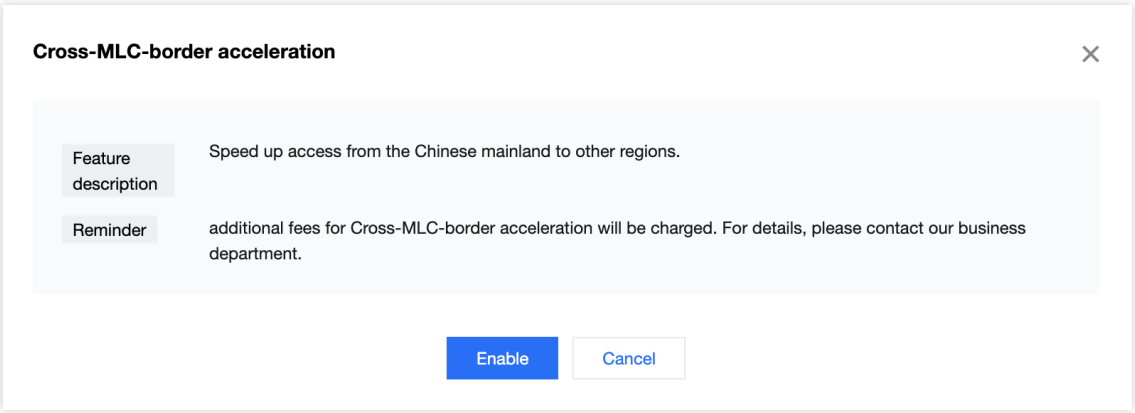
1. This function is only supported by the EdgeOne Enterprise plan and is currently in beta testing. Please contact business support if you need to enable it.
2. Additional fees for the Cross-MLC-border acceleration function will be charged. For details, please contact our business department.

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, click the site list in the left menu bar, and click the site you want to configure.
2. On the site details page, click Site Acceleration > 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.

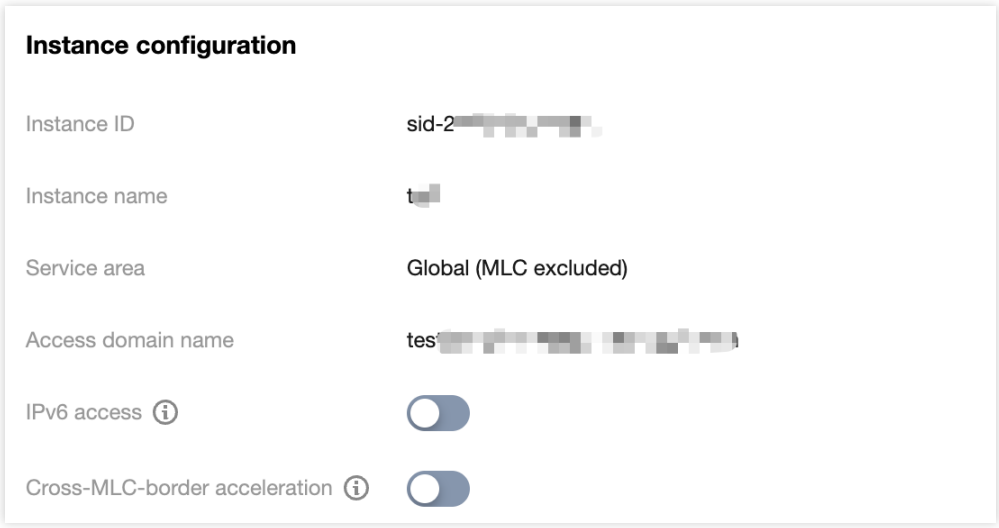


4. In the confirmation window, click **Enable** to complete the configuration.

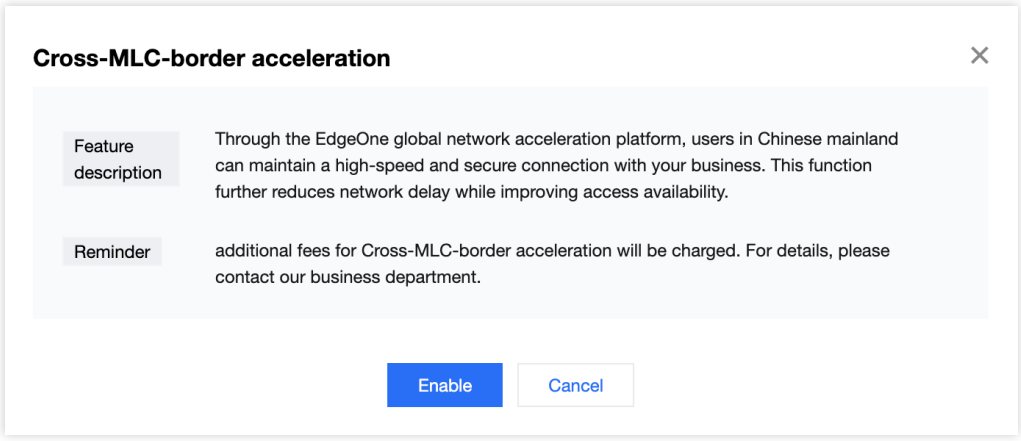


If you need to enable the Cross-MLC-border acceleration function for a single L4 proxy instance, please follow the steps below:

1. Log in to the EdgeOne console, click the site list in the left menu bar, and click the site you want to configure.
2. On the site details page, click L4 Proxy.
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:

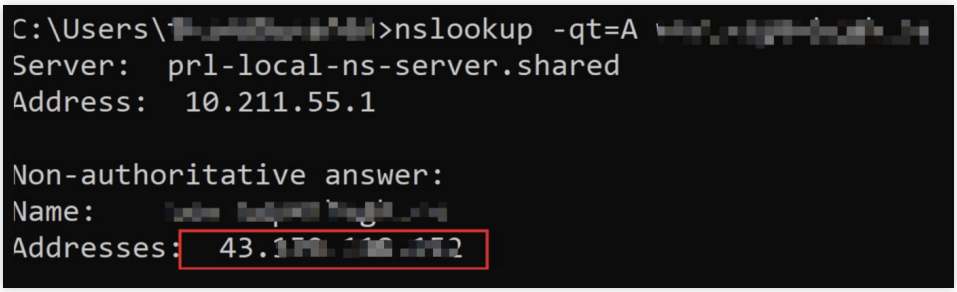
Note： Please ensure that the access test is initiated from the Chinese Mainland since the Cross-MLC-border acceleration function affects the outgoing user requests from the Chinese Mainland.

Windows

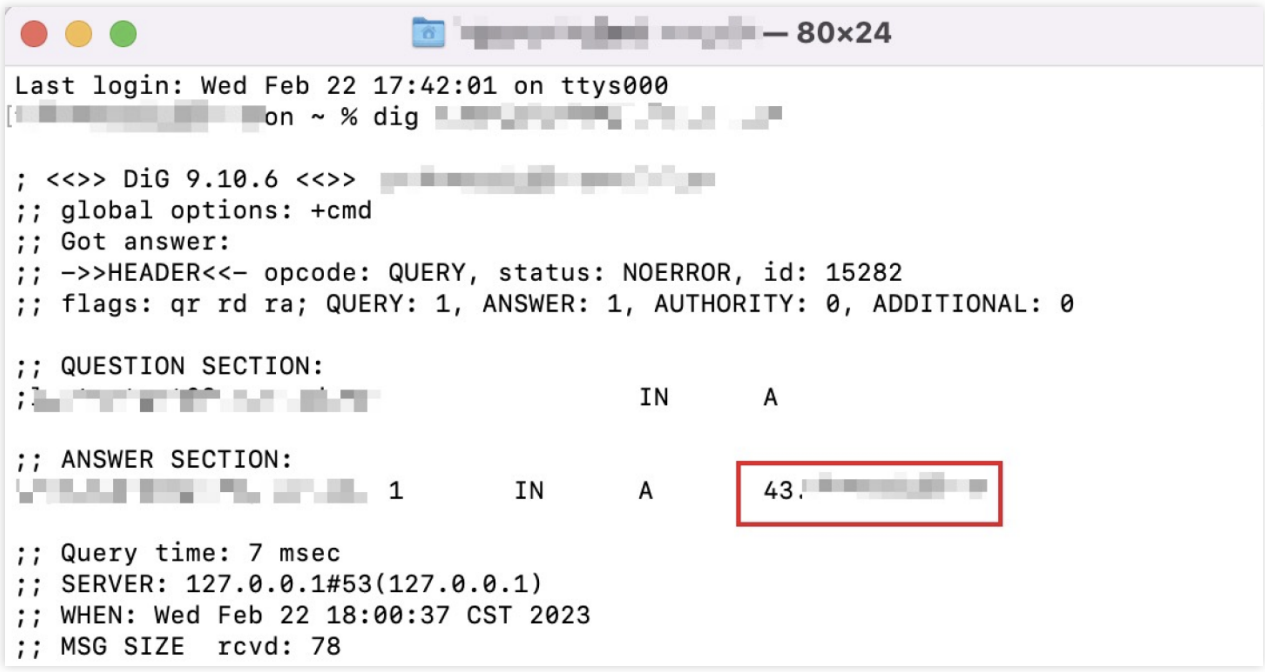
Mac/Linux

Visit the site

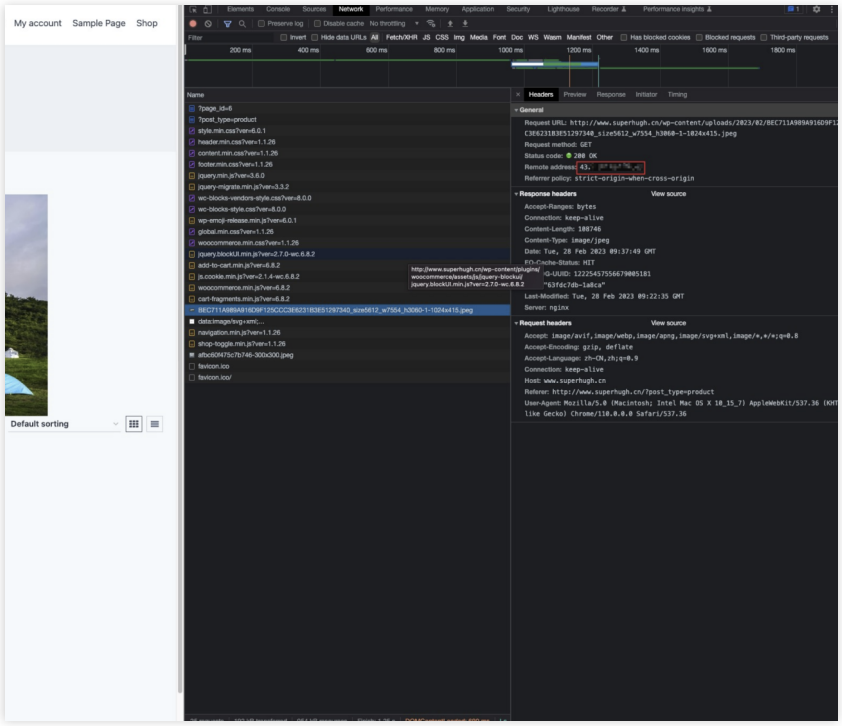
In Windows system, open the command prompt. Taking the domain `www.example.com` as an example, run the `nslookup -qt=A www.example.com` command. Then you can get the IP address of the domain obtained by the A record resolution.



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.



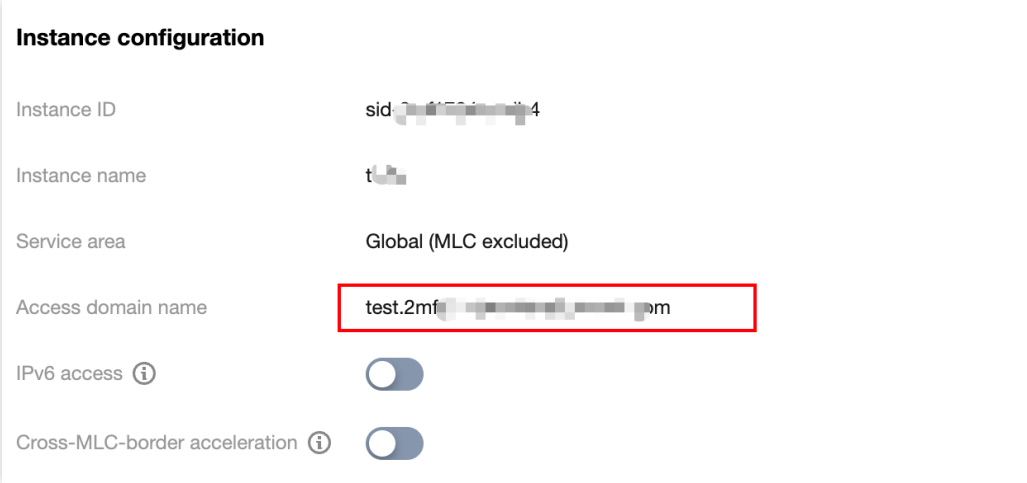
You can also obtain the IP address by visiting the site. Taking the domain `www.example.com` as an example, you can press F12 in the browser to open the developer tools. Then click any request record, and you can view the IP address that the request points to.



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.



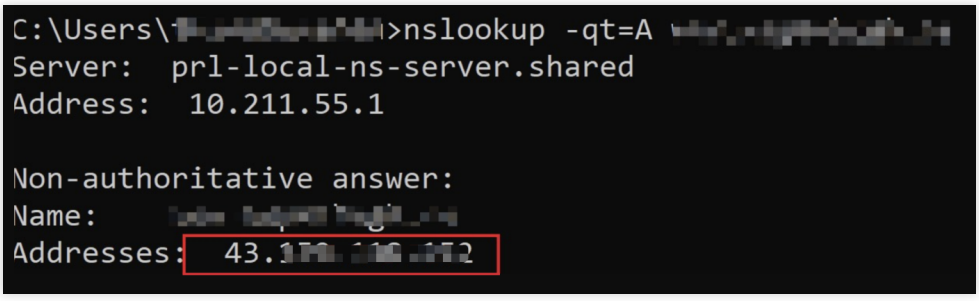
2. You can obtain the IP address of the assigned node by using any of the following methods:

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,

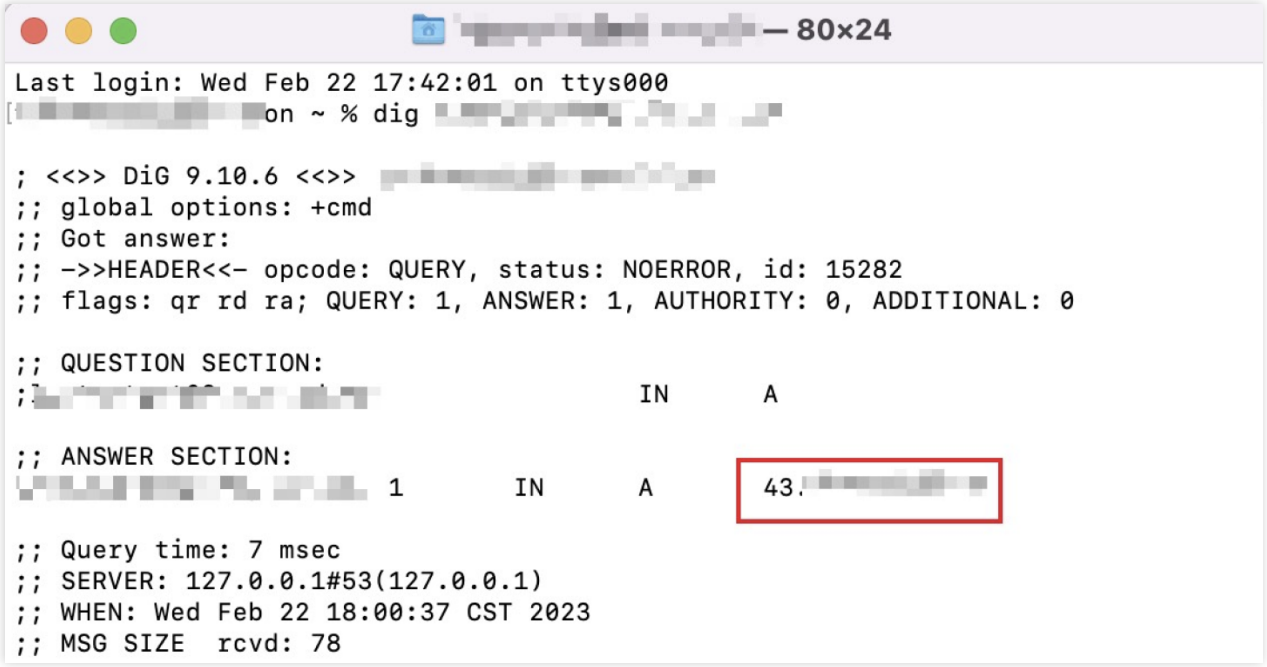
Windows

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.

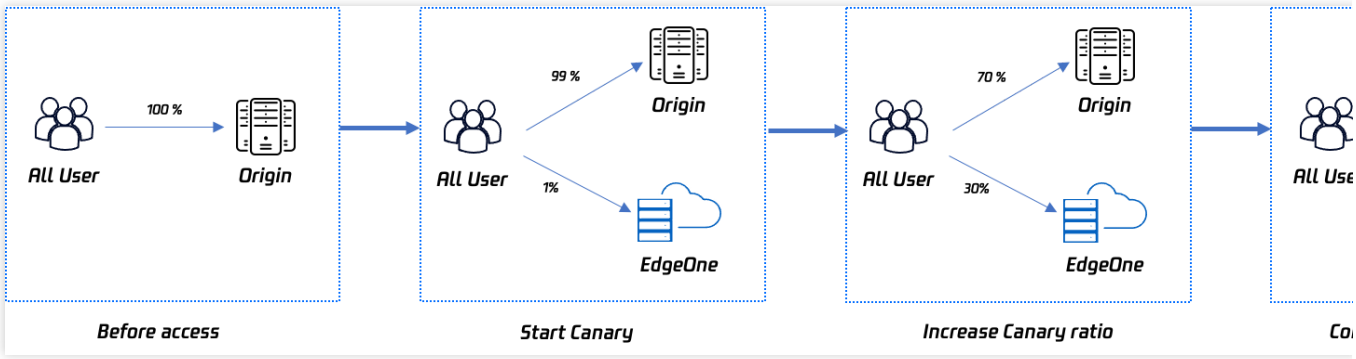


3. You can query the location information of the IP through any IP query tool. If it belongs to Tencent Hong Kong, the Chinese mainland network optimization (international acceleration) function is effective.

Scheduling Traffic 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.

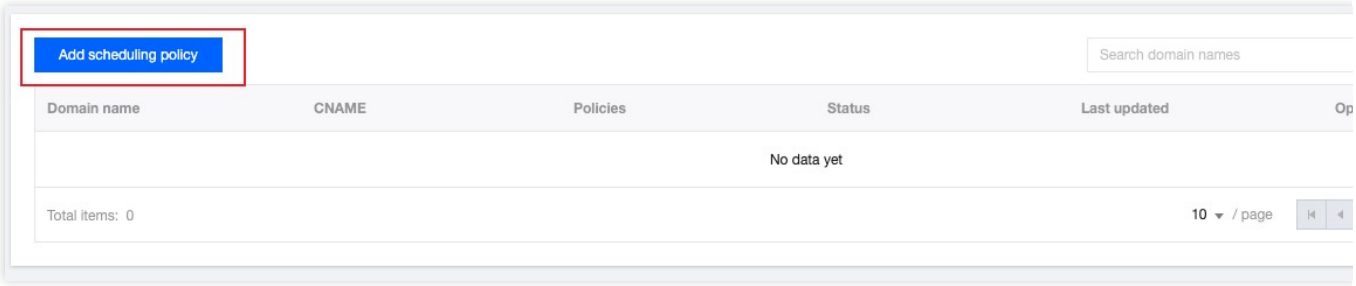
Use Cases

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.

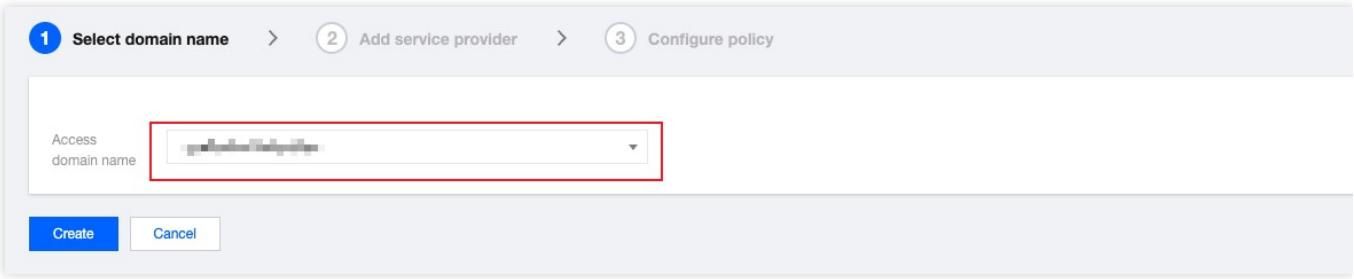
Directions

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**.

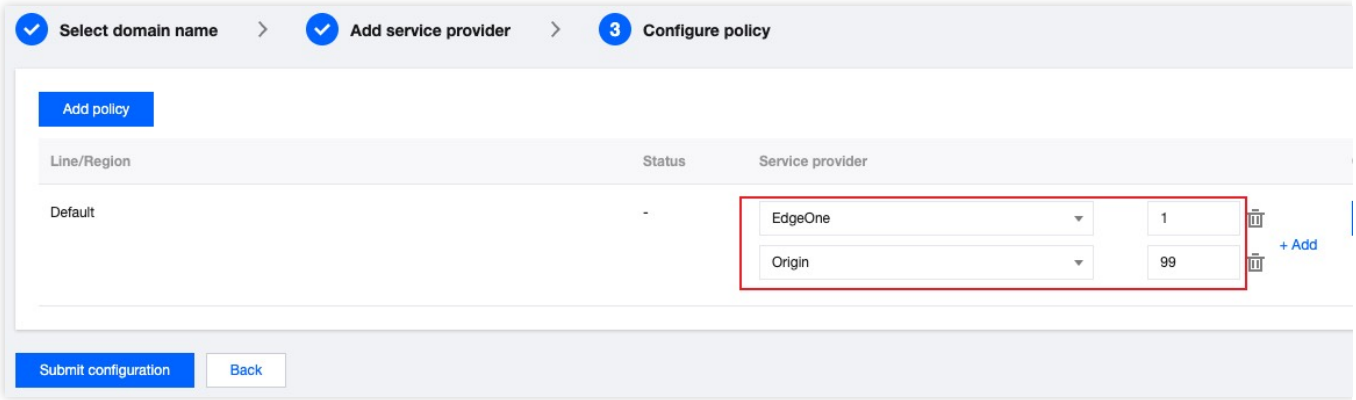


3. In the **Select domain name** step, select `huidu.example.com` from the Access domain name drop-down list and click **Create**.



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**.

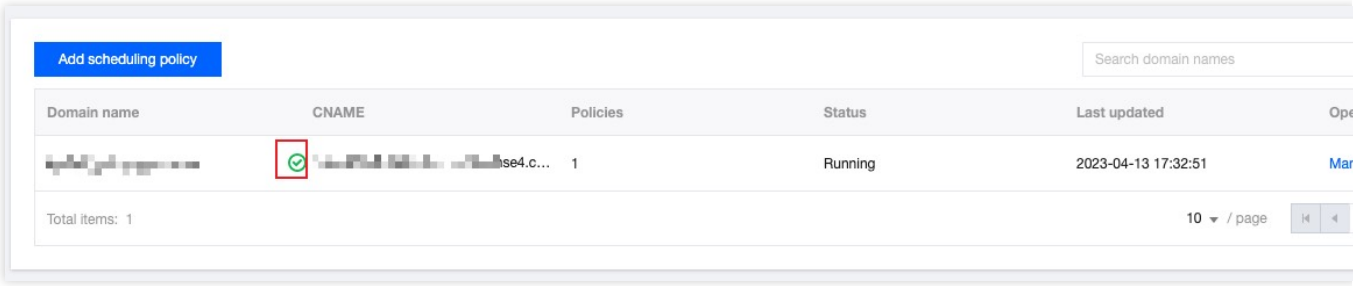
5. In the **Configure policy** step, add an initial canary switching policy and click **Submit**. Set the weight of the service provider `origin domain name` to `99` and that of EdgeOne to `1`. This policy means to switch 1% of traffic from the origin to EdgeOne. You can increase the traffic migration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

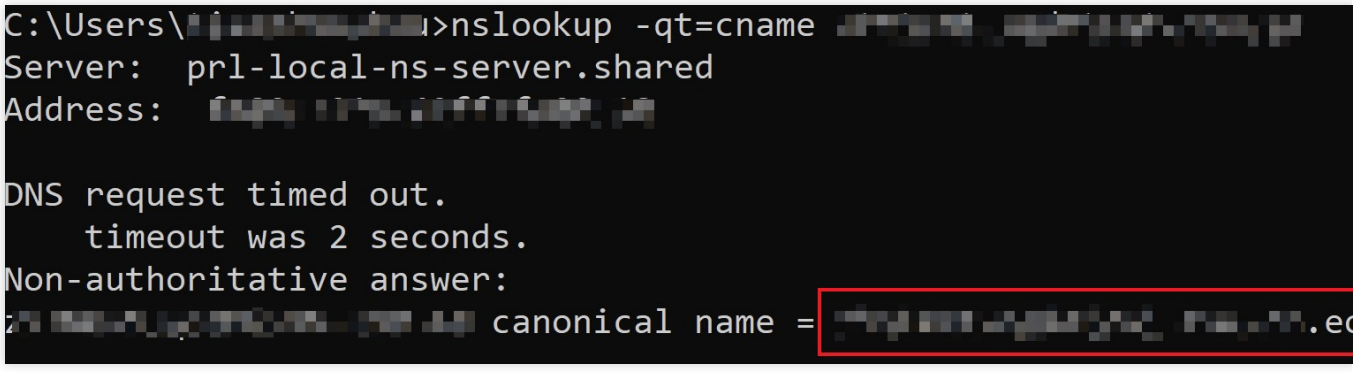
You can run the `nslookup` or `dig` command to check the switching result.

Windows

macOS or Linux

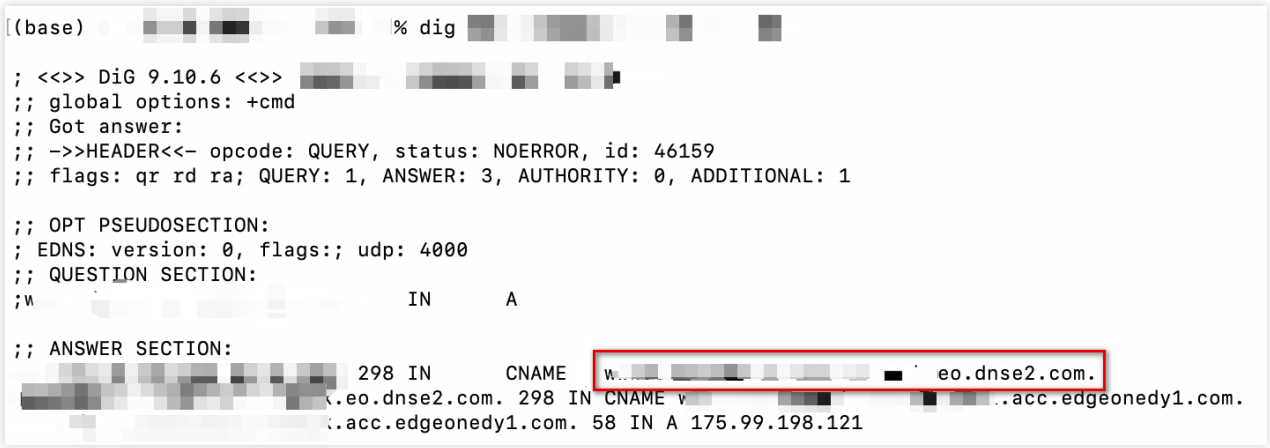
Open the command prompt and run `nslookup -qt=cname huidu.example.com` . Then, check the ratio of the CNAME addresses in the DNS result.

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.



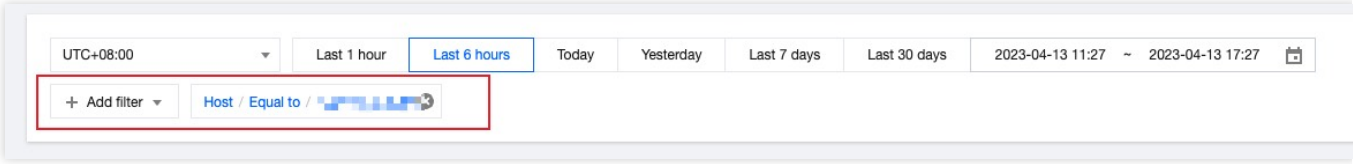
Open the terminal and run `dig huidu.example.com` . Then, check the ratio of the CNAME addresses in the DNS result.

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.



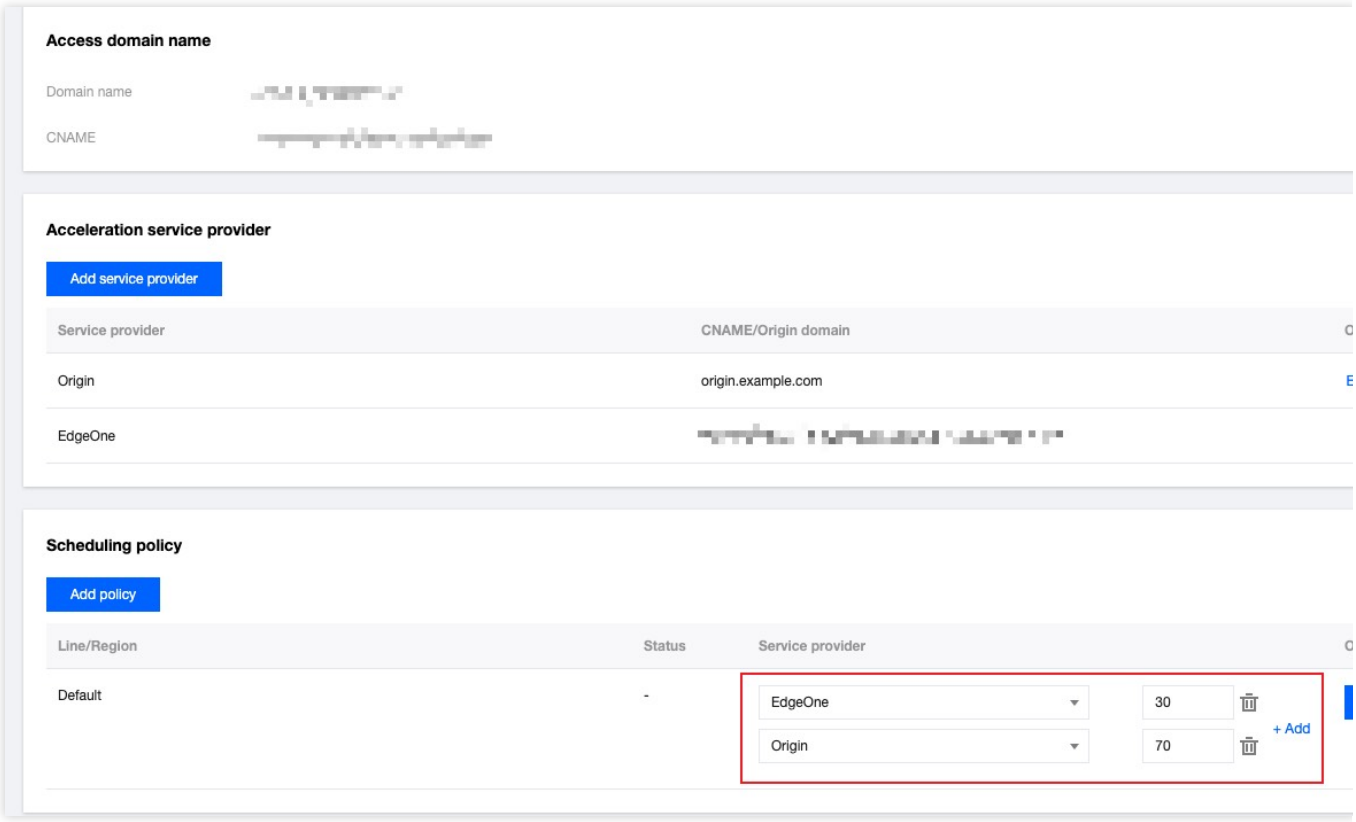
3. Viewing traffic changes

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.



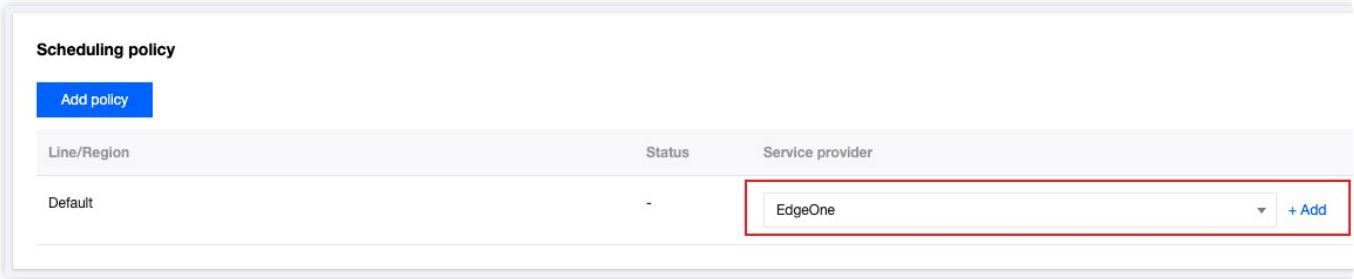
Step 3. Increase the traffic migration ratio

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.



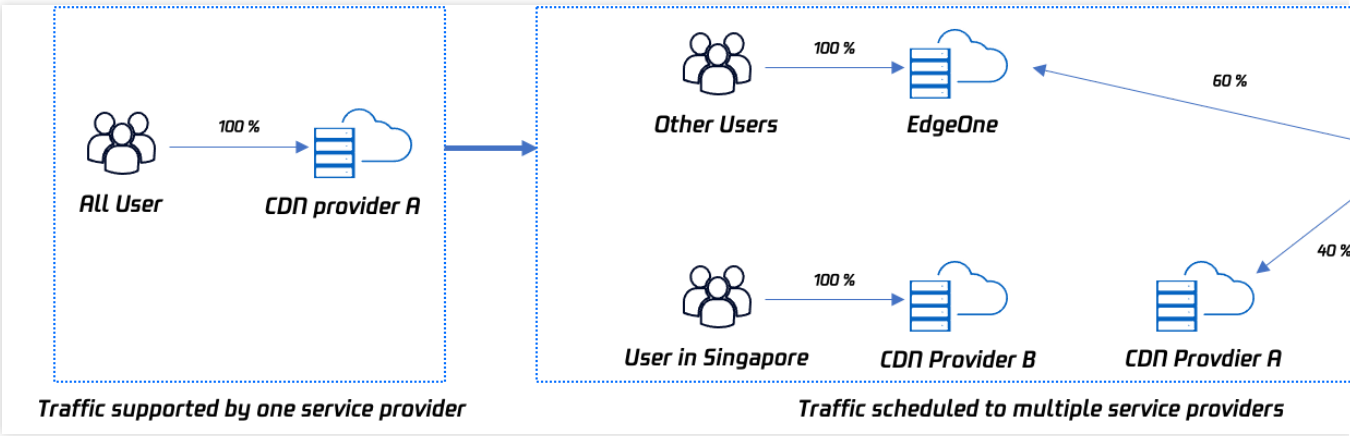
2. You can disable and delete the traffic scheduling policy later if the service remains stable after 100% canary switching. At this point, disabling or deleting the policy has no impact on the service, and the traffic is fully managed by EdgeOne.

Relevant Documentation

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.

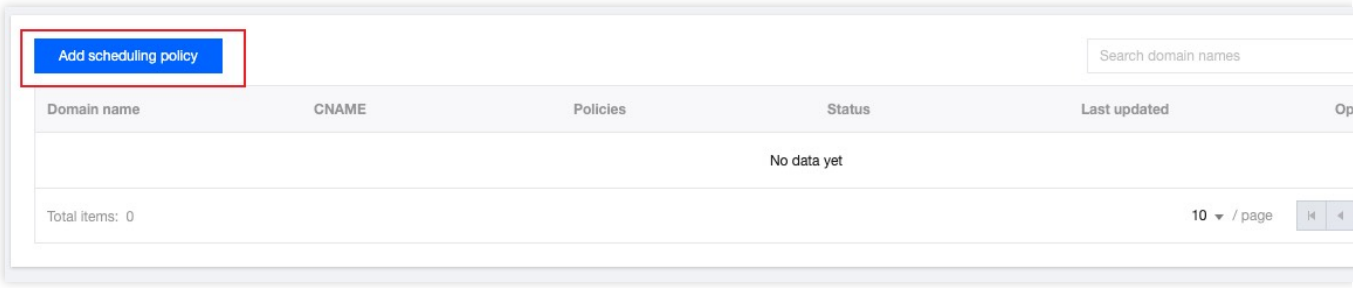
Preset Scenarios

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.

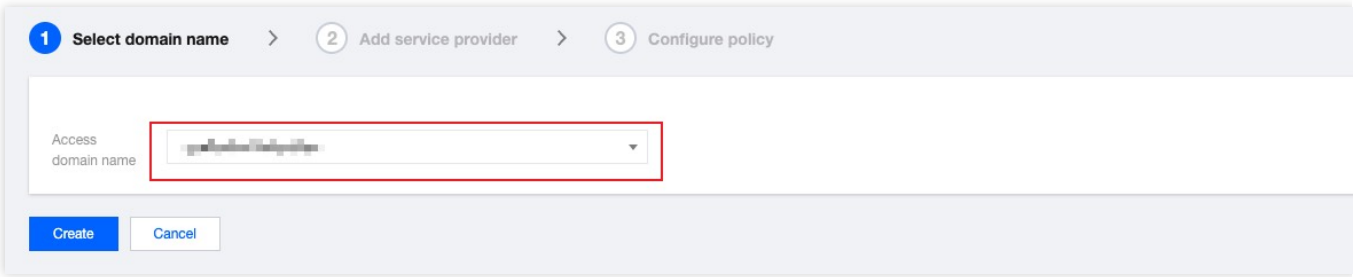
Operation Steps

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**.

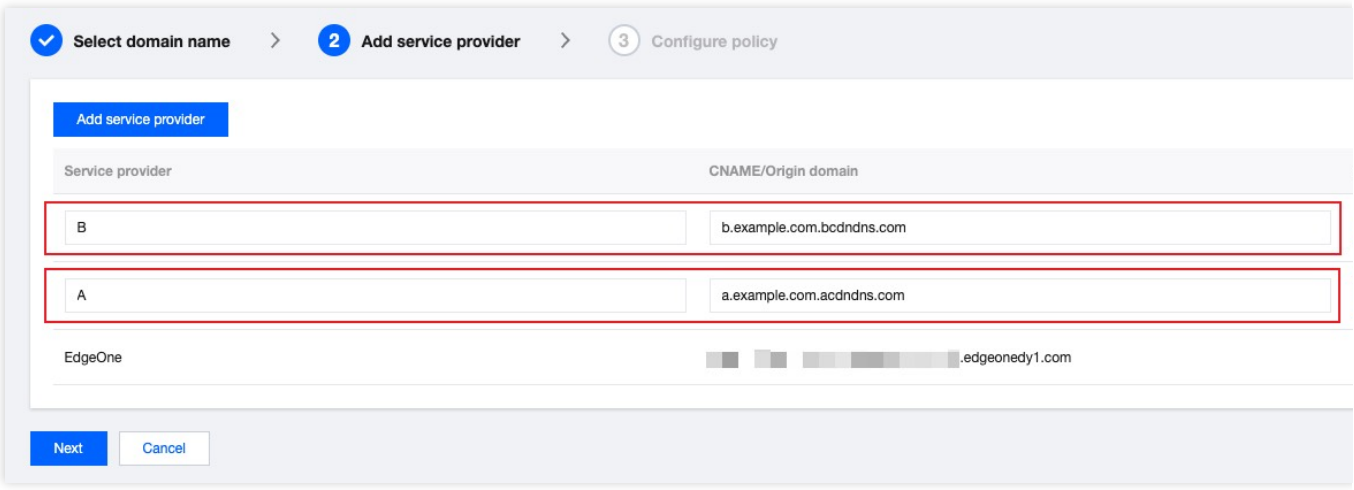


3. On the Traffic Scheduling Management page, click **Add Scheduling Policy**, select `a.example.com` , and click **Create**.



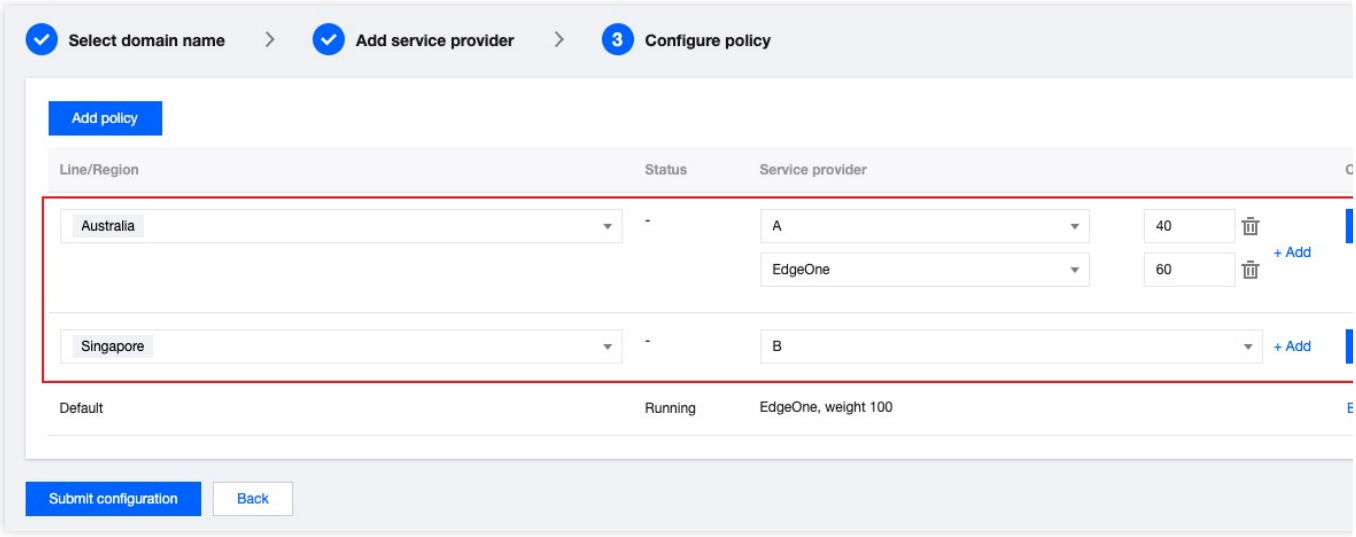
Step 2: Set the policy

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.



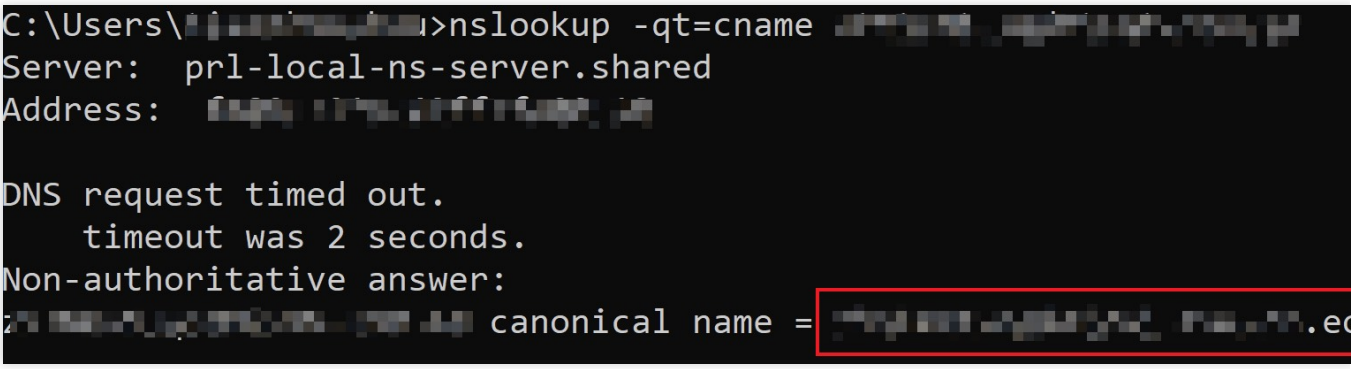
Step 3: Switch resolution

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.

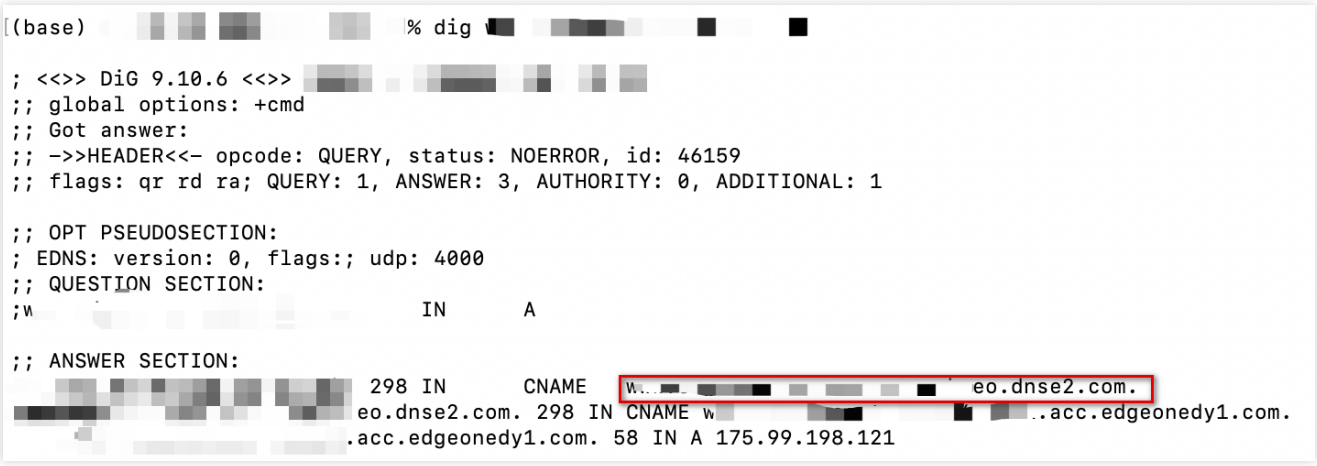
Step 4: Verify Effectiveness

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.

EdgeOne facilitate APKs.s 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

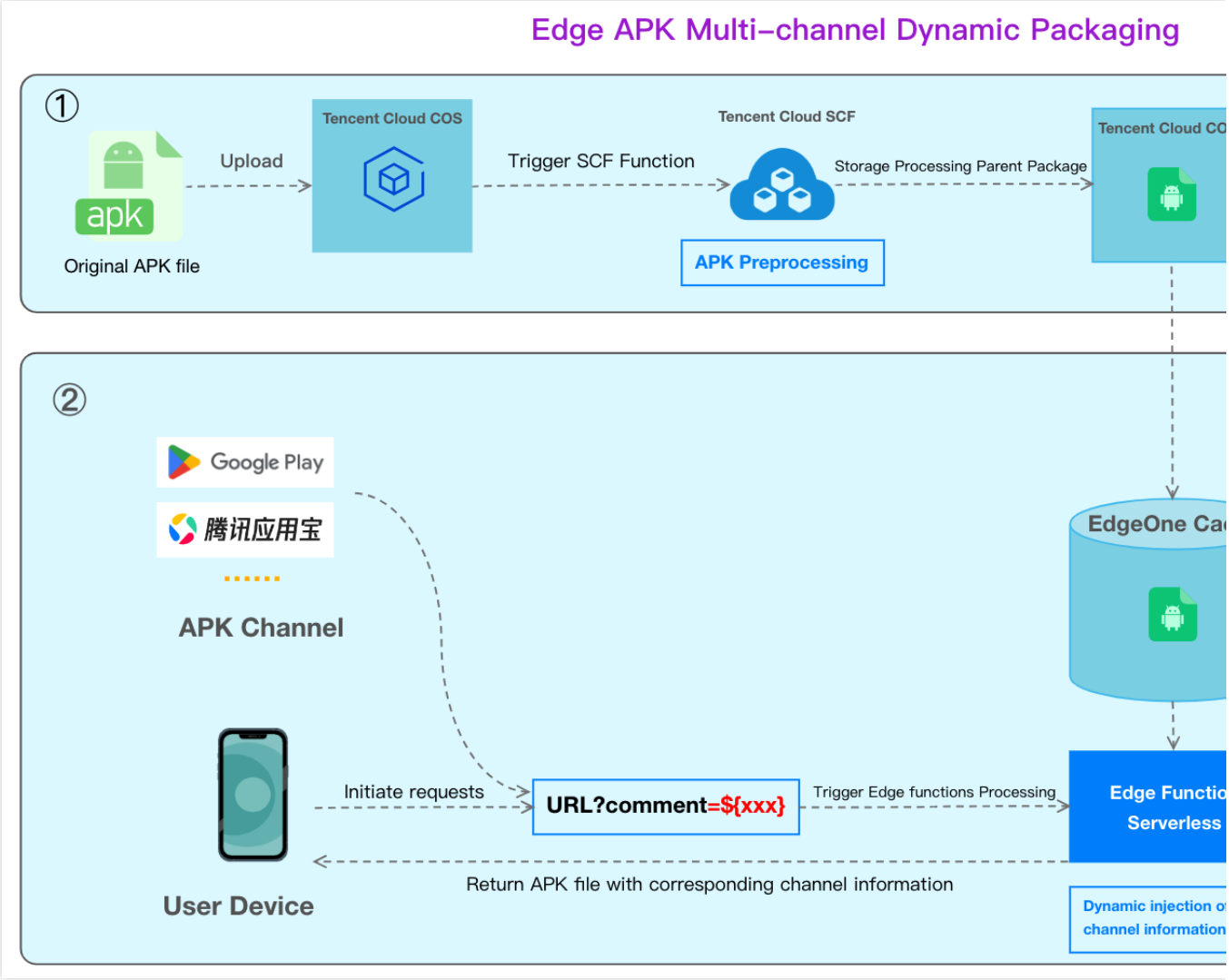
APK (Android Application Package) is the installation package for Android applications. When an app releases a new version, it typically requires the creation of distinct channel installation packages for each distribution channel. These packages are then uploaded to the respective application markets. 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

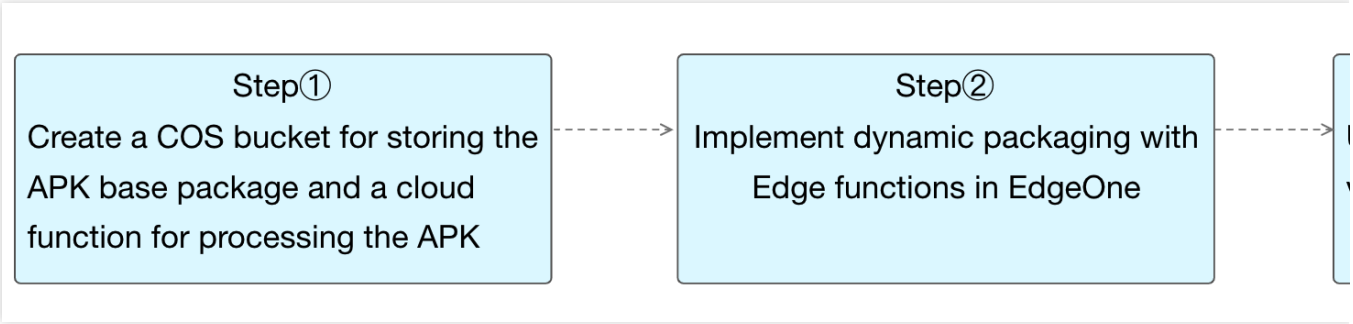
1. Reduced Channel Package Maintenance Costs: Developers only need to maintain an original Android APK parent package, eliminating the need to manage individual packages for each channel partner. EdgeOne provides default packaging tools, allowing users to deploy easily through simple UI configuration 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	
Packing speed	Fast	Fast	Faster	Ne de an sig
Channel information injection form	Dynamic	Static	Static	
Channel information injection side	Edge	Origin	Origin	
APK output quantity	One	Multiple	Multiple	
packing & accleration	EdgeOne one-stop packing & acceleration	Not support acceleration	Not support acceleration	

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

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).

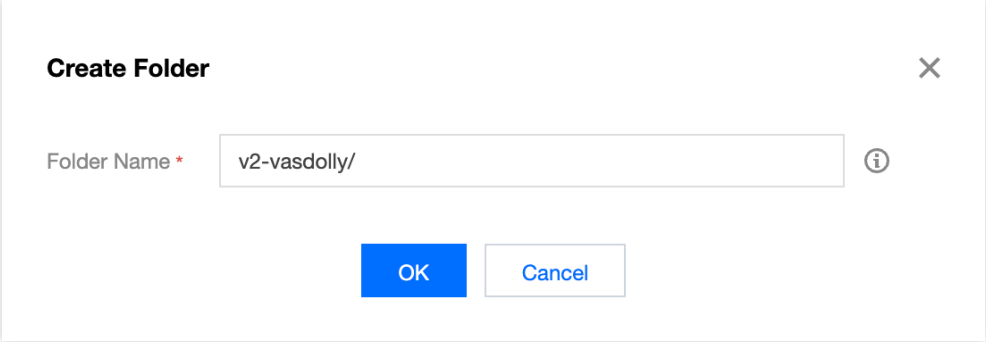
Preparation

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.

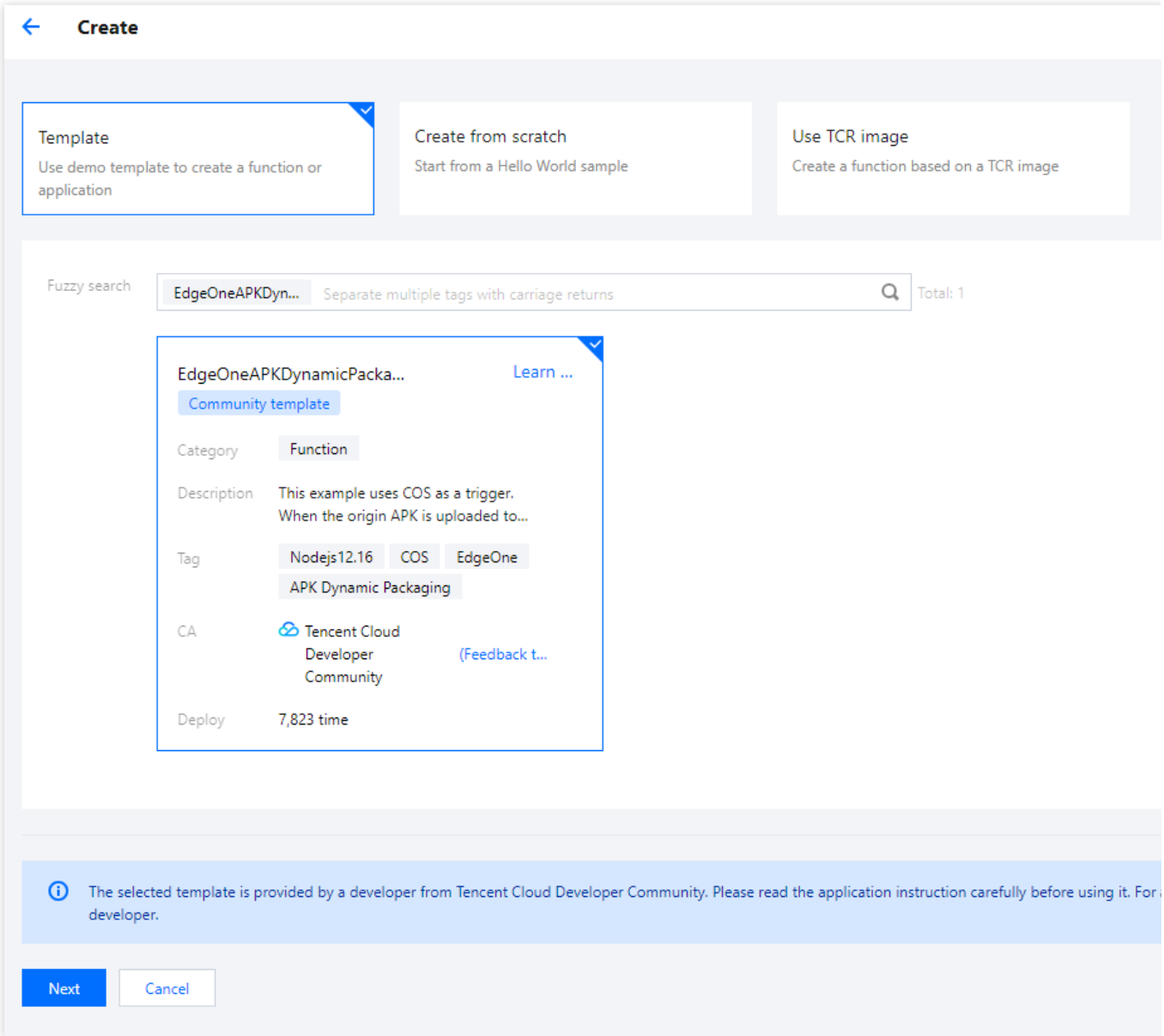


Step 2: Create a New Template Function

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:

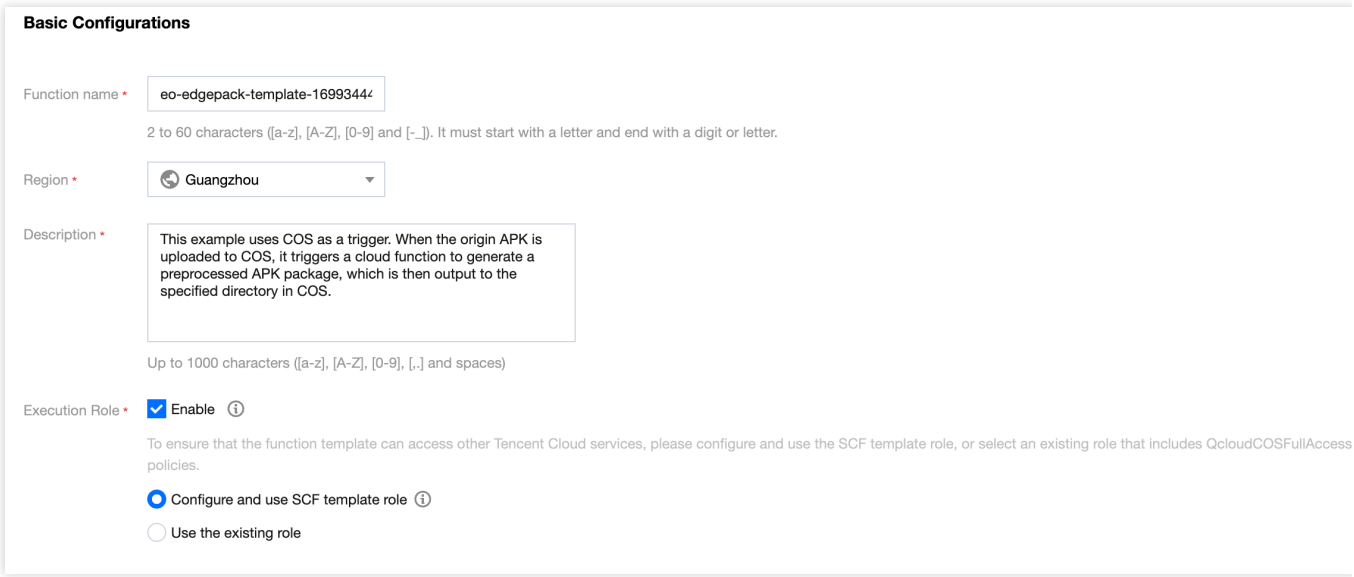
Basic Configuration

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:

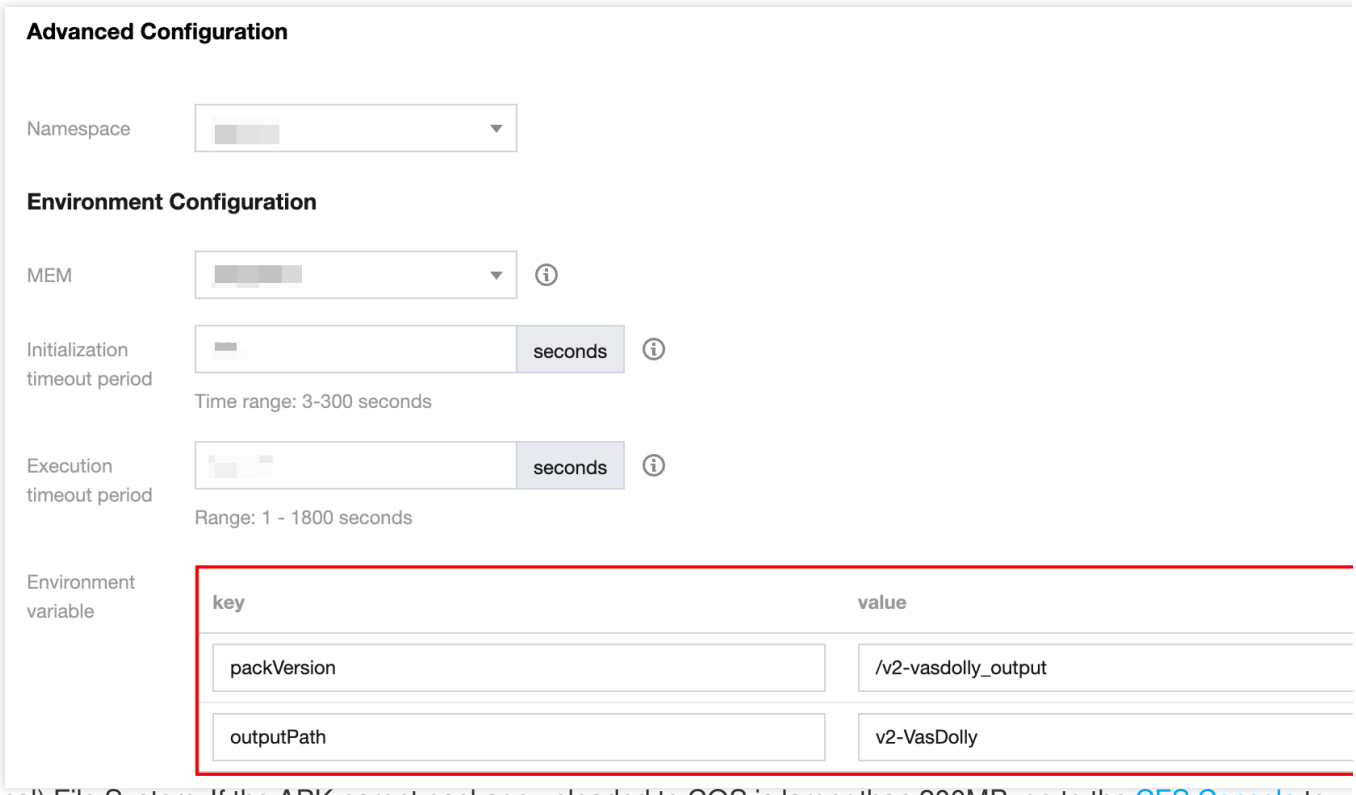
outputPath (Required): Customize the directory in the COS bucket where the Cloud Function SCF outputs the processed APK parent package, for example, `/v2-vasdolly_output` .

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

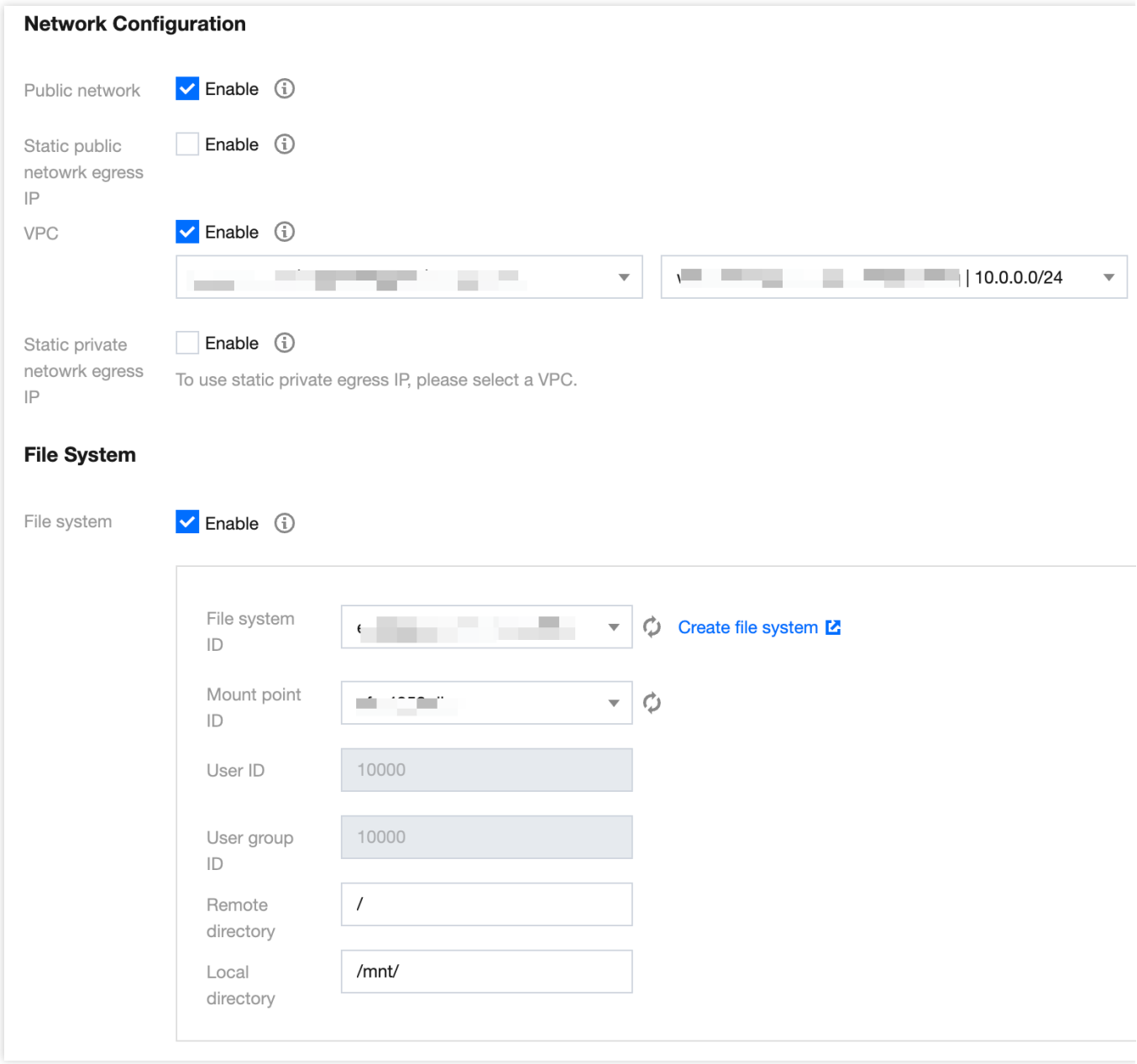
APK Signature Version	packVersion Value
v1	v1
v2	Please enter v2-VasDolly, v2-Walle, or v2-Custom: v2-VasDolly: Store the channel information in the ID-Value pair with the ID `0x881155ff` (VasDolly default). v2-Walle: Store the channel information in the ID-Value pair with the ID `0x71777777` (Walle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

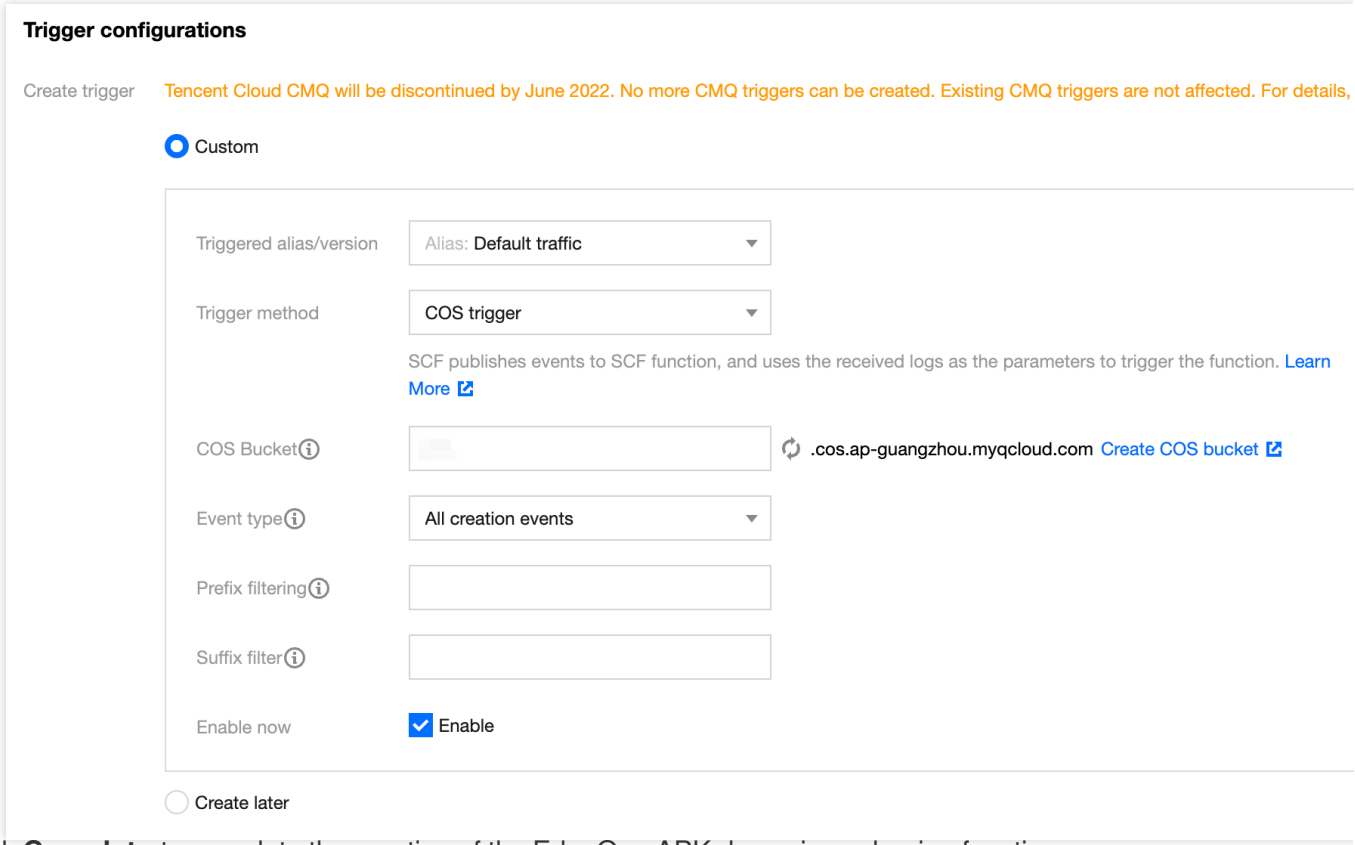
Trigger Management

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：2023-12-05 17:53:15

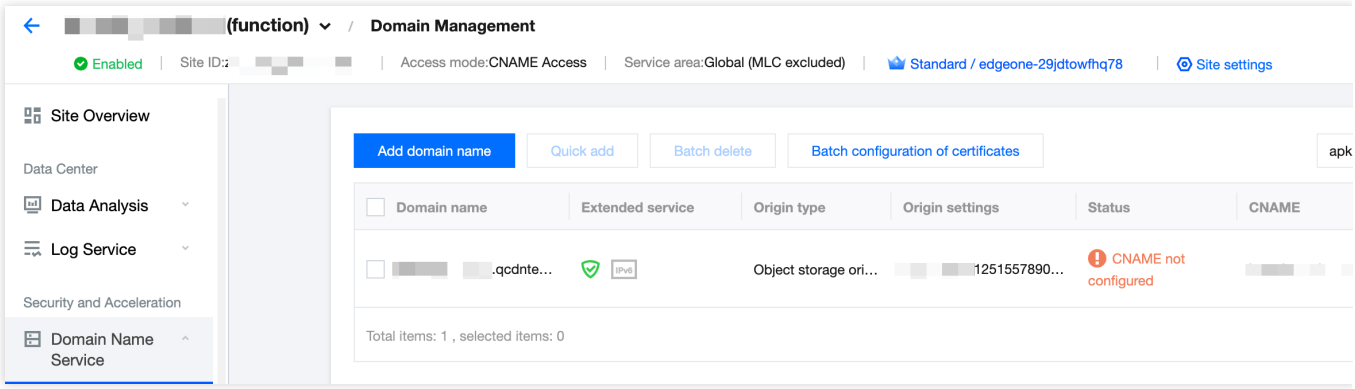
Through EdgeOne edge function, we can dynamically write channel information into the APK package. Users only need to access the domain bound to the edge function and trigger the appropriate configuration to enable the edge function, achieving dynamic packaging and accelerated distribution of the APK.

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:

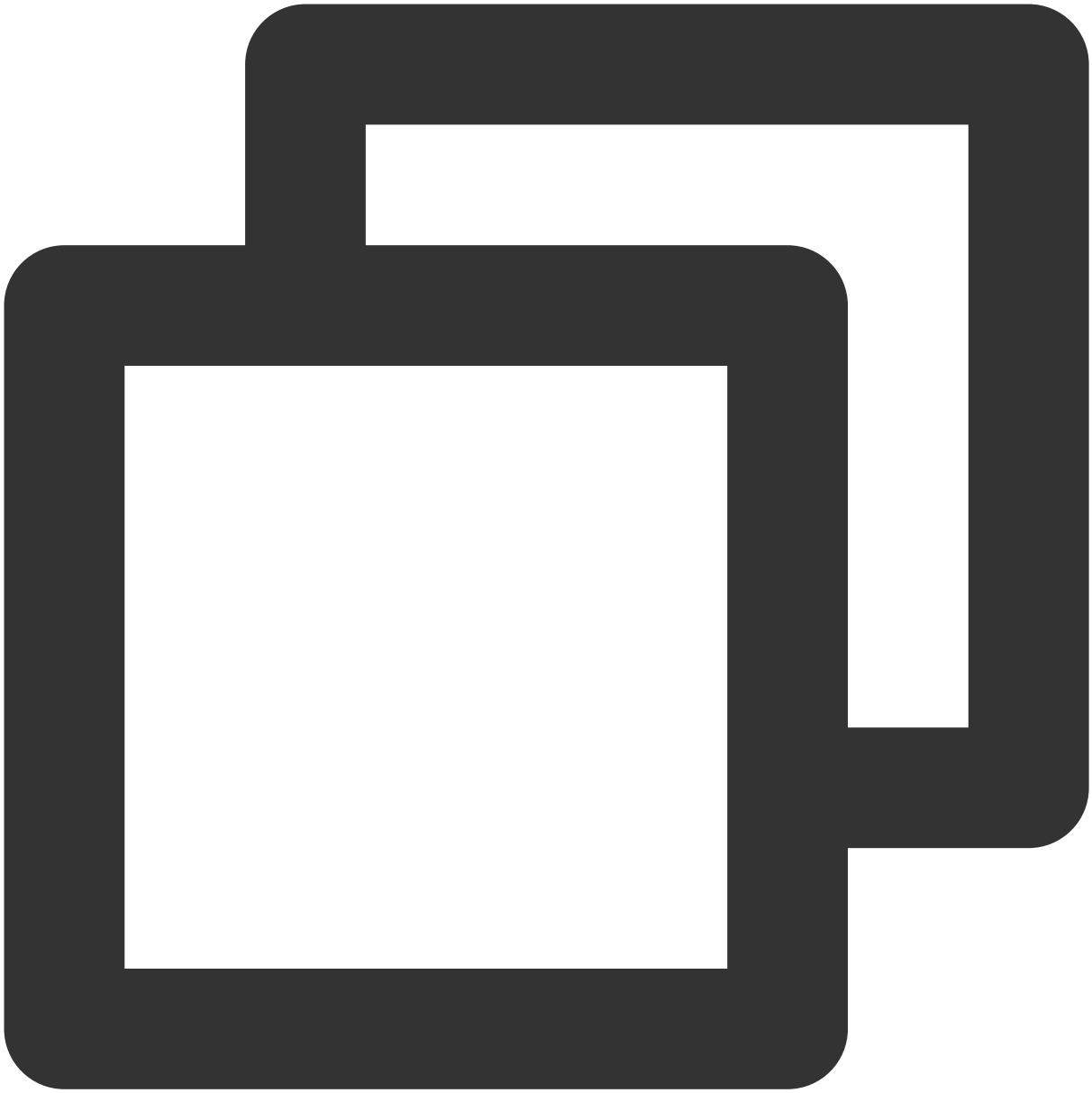
Note:

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;

    /** 1. Preliminary verification of the request. Any request that doesn't requir
    if (!this.checkRequest(request)) {
      return;
    }

    /** 2. Utilize fetch to retrieve the source file */
    let response = null;
    try {
      response = await fetch(request);
    } catch (err) {
      const error = {
        code: 'FETCH_ORIGIN_ERROR',
        message: err?.message,
      };
      response = new Response(JSON.stringify(error), {
        status: 590,
      });
    }

    /** 3. Verification of the response. A response that doesn't require handling,
    if (!this.checkResponse(response)) {
      return event.respondWith(response);
    }

    /** 4. Manage the APK file and respond to the client */
    const { readable, writable } = new TransformStream();
    this.handleStream(response, writable);

    response.headers.set('Cache-Control', 'max-age=0');
    const streamResponse = new Response(readable, response);
```

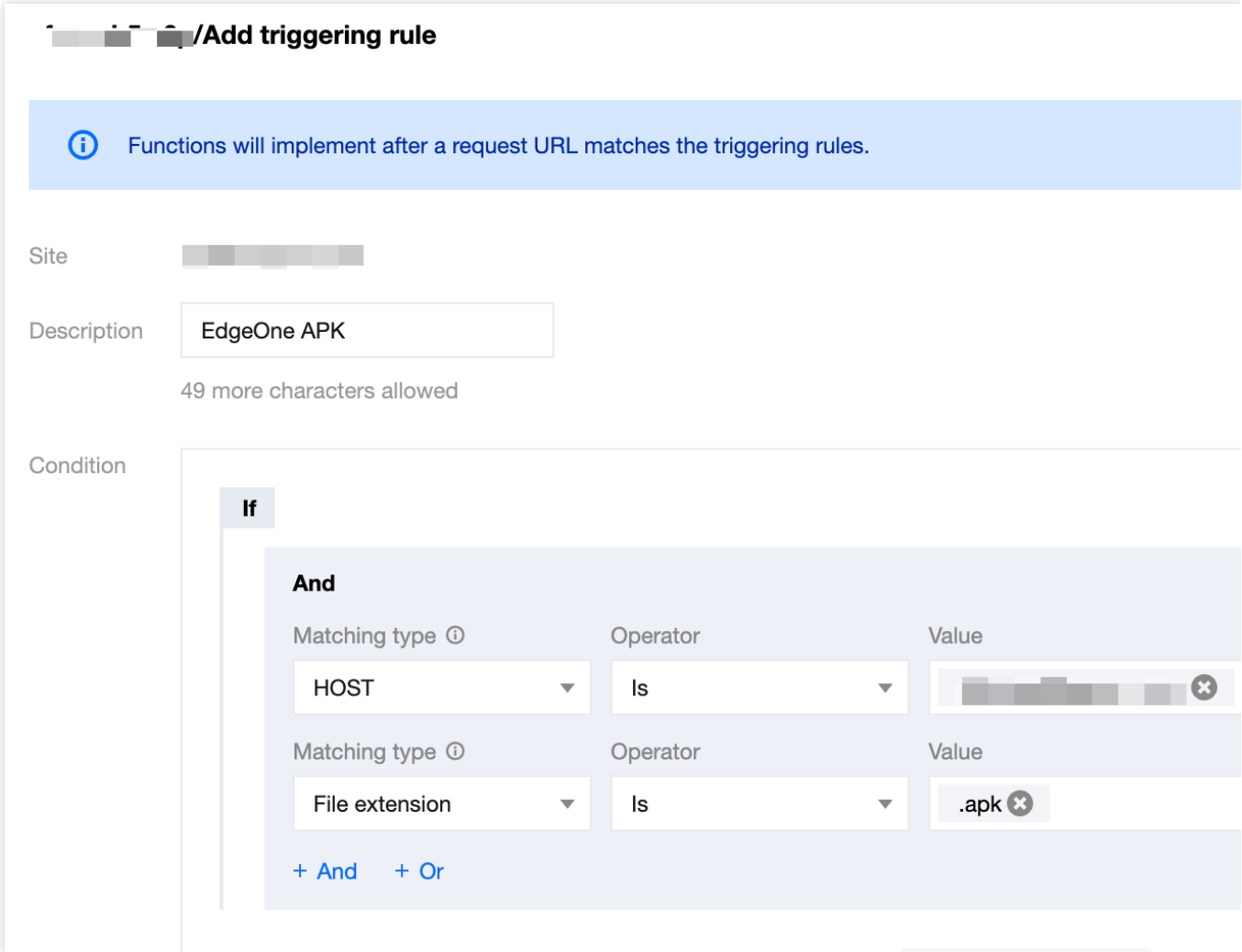
```
    event.respondWith(streamResponse);
  }

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

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

    /** ATTENTION: By default, the 'comment' parameter is taken, should there be a
    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;
  }

  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];
  }

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

    const responseBody = response.body;
    const encoder = new TextEncoder();

    const section = encoder.encode(comment);
```

```
    const writer = writable.getWriter();
    const reader = responseBody.getReader();

    try {
      let handledBytes = 0;
      while (true) {
        const result = await reader.read();

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

        await writer.ready;
        await writer.write(buffer);
      }
    } catch (err) {
      console.error('WRITE_COMMENT_ERROR: ', err);
    }

    try {
      await writer.ready;
      await writer.close();
    } catch (err) {
      console.error('CLOSE_WRITER_ERROR: ', err);
    } finally {
      writer.releaseLock();
    }
  }
}
```

```
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:

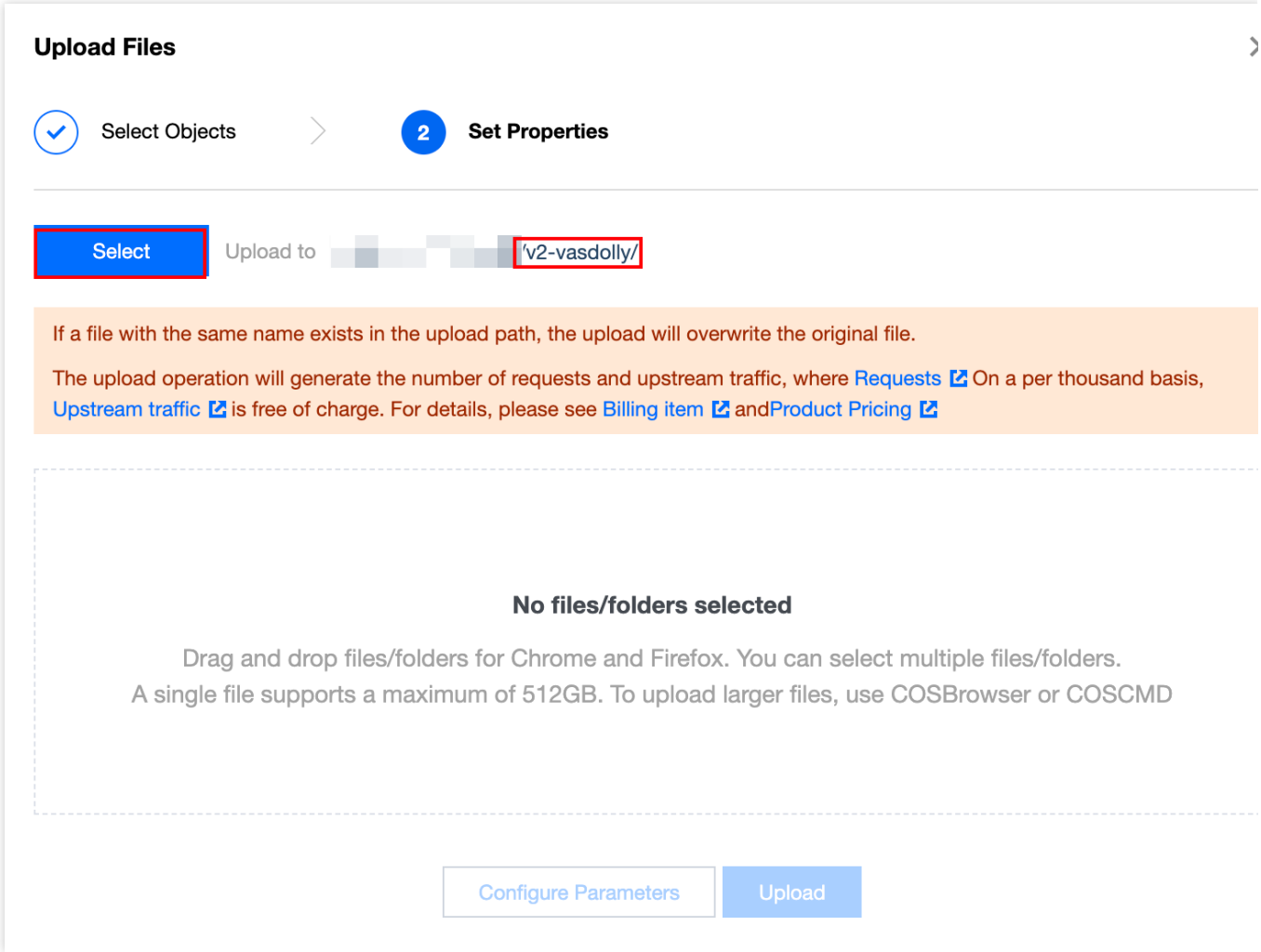
Please proceed to Step 3:Implement Test and Verify the Outcome Effectiveness.

Step 3: Implement Test and Verify the Outcome Effectiveness

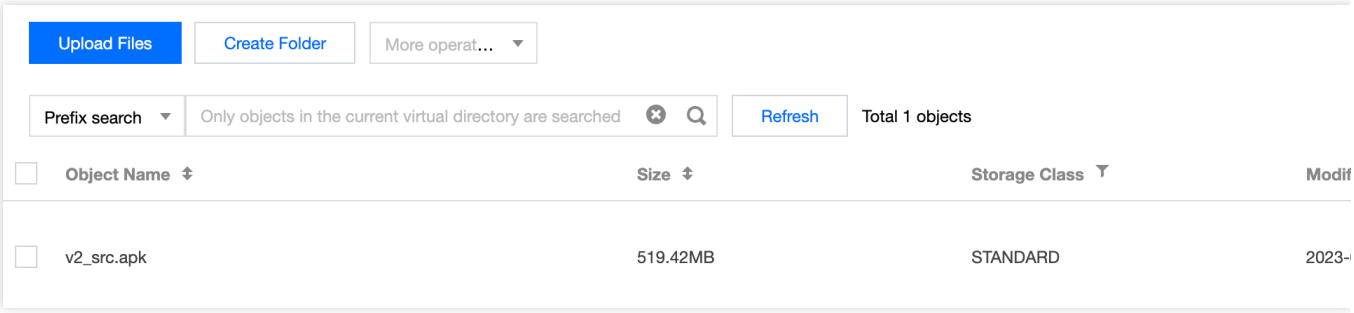
Last updated：2023-12-05 17:53:33

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.