ALIBABA CLOUD

Alibaba Cloud

dcdn

Domain Management

Document Version: 20220712

Alibaba Cloud

Legal disclaimer

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

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

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

Table of Contents

1.Feature 07
2.Copy configurations to domain names 13
3.Verify the ownership of a domain name 15
4.Tag management 19
4.1. Tag overview 19
4.2. Tag management 20
5.Basic Settings 23
5.1. Overview 23
5.2. Modify basic information 23
5.3. Configure an origin server 24
6.Origin Fetch Settings 26
6.1. Overview 26
6.2. Configure an origin host 27
6.3. Configure the static origin protocol policy 29
6.4. Configure private bucket origin 31
6.5. Configure an origin SNI 34
6.6. Configure a common name whitelist 35
6.7. Configure range origin fetch 36
6.8. Configure a timeout period for origin fetch requests 37
6.9. Customize an origin HTTP header 38
6.10. Rewrite URIs in back-to-origin requests 39
7.Dynamic and static acceleration rules 43
7.1. Overview 43
7.2. Configure static file types 43
7.3. Specify static file URIs 46
7.4. Specify static file paths 47

7.5. Configure back-to-origin requests to retrieve dynamic cont... 48
8.Cache settings 51
8.1. Overview 51
8.2. Add a cache rule for resources 51
8.3. Create a cache expiration rule for HTTP status codes 58
8.4. Customize an HTTP header 61
8.5. Customize an error page 63
8.6. Create a URI rewrite rule 64
9.HTTPS settings 67
9.1. What is HTTPS secure acceleration? 67
9.2. Certificate formats 72
9.3. Configure an SSL certificate 74
9.4. Enable HTTP/2 77
9.5. Configure OCSP stapling 78
9.6. Configure force redirect 79
9.7. Configure TLS version control 81
9.8. Configure HSTS 82
9.9. Enable ShangMi for HTTPS 84
9.10. Enable authentication on client certificates 85
10.Access Control 87
10.1. Overview 87
10.2. Configure a referer whitelist or blacklist to enable hotlin... 87
10.3. URL authentication 90
10.3.1. Configure URL authentication 90
10.3.2. Type A signing 95
10.3.3. Authentication type B 97
10.3.4. Authentication type C 98
10.4. Configure an IP address blacklist or whitelist 99

10.5. Configure a User-Agent blacklist or whitelist 101
11.Performance Optimization 103
11.1. Overview 103
11.2. Configure HTML optimization 103
11.3. Configure intelligent compression 104
11.4. Configure Brotli compression 105
11.5. Configure image editing 106
11.5.1. Image editing and its benefits 106
11.5.2. Convert image formats 107
11.5.3. Change image quality 108
11.5.4. Crop images 108
11.5.5. Resize images 108
11.5.6. Rotate images 109
11.5.7. Change the color of an image 109
11.5.8. Manage image watermarks 109
11.5.9. Query image information 111
11.6. Configure the parameter filtering feature 112
11.7. Configure video seeking 116
12.Security Settings 118
12.1. Configure bot traffic management 118
12.2. Configure precise access control 119
13.Advanced Settings 123
13.1. Configure IPv6 123
14.WebSocket 124
14.1. What is WebSocket? 124
14.2. Configure WebSocket 125

1.Feature

The Alibaba Cloud Dynamic Route for CDN (DCDN) console allows you to configure and manage domain names. It also provides resource monitoring for real-time data analysis. The DCDN console also displays the billing status and allows you to change the billing method. This topic describes the DCDN console and the domain management features.

> **Note** To help you understand and obtain up-to-date information about DCDN, this topic divides the features in the DCDN console into domain management and service management based on your business requirements.

Domain management features

Feature	Reference	Description	Default
Copy configurations	Copy configurations to domain names	Allows you to copy one or more configurations of an accelerated domain to another one or more domains.	None
Basic settings	Modify basic information	Allows you to modify the accelerated region.	None
	Configure an origin server	Allows you to modify the origin information.	None
	Configure an origin host	Allows you to modify the domain name of the origin host.	Disabled
	Configure the static origin protocol policy	Enables DCDN to communicate with the origin based on the specified origin protocol policy. If you specify the Match Client policy, DCDN communicates with the origin over HTTP or HTTPS, depending on the protocol of the client request.	Disabled
	Configure private bucket origin	Grants DCDN permissions to access the specified private Object Storage Service (OSS) bucket that serves as the origin.	Disabled

Feature	Back-to-origin settings	Reference	Description	Default
		Configure an origin SNI	Allows you to set a Server Name Indication (SNI) value to specify the requested domain name when DCDN communicates with the origin over HTTPS. You must enable this feature if the origin IP address is bound to multiple domain names.	Disabled
		Configure range origin fetch	Enables DCDN to retrieve content from the origin based on HTTP range requests. This reduces the back-to-origin data usage and shortens the resource response time.	Disabled
		Customize an origin HTTP header	Allows you to add or remove HTTP headers when DCDN communicates with the origin over HTTP.	Disabled
Dynamic acceleration rules		Specify static file paths	Allows you to specify the paths of the static files.	Disabled
		Configure static file types	Allows you to specify the file extensions of the static files.	Disabled
		Specify static file URIs	Specifies the Uniform Resource Identifiers (URIs) of the static files.	Disabled
		Configure back-to-origin requests to retrieve dynamic content	Enables DCDN to communicate with the origin based on the specified origin protocol policy when DCDN requests dynamic content from the origin. If you specify the Match Client policy, DCDN communicates with your origin over HTTP or HTTPS, depending on the protocol of the client request.	Disabled

Feature	Reference	Description	Default
Cache settings	Add a cache rule for resources	Allows you to customize cache expiration rules for specified resources.	None
	Customize an HTTP header	Allows you to customize HTTP response headers. DCDN provides 10 HTTP response headers for customization.	None
	Customize an error page	Allows you to customize a full URL to redirect to for an HTTP or HTTPS response code.	404
	Create a URI rewrite rule	Allows you to modify a request URI and perform a 302 redirect to the specified destination URI.	None
	Configure an SSL certificate	Provides an end-to-end HTTPS secure acceleration solution. You can enable the secure acceleration mode and upload the certificate and the private key for an accelerated domain. This feature also allows you to view, disable, enable, or modify the certificate.	Disabled
	Enable HTTP/2	Enables the binary protocol HTTP/2 to provide multiple benefits including extensibility, security, multiplexing, and header compression.	Disabled

Feature	Reference	Description	Default
HTTPS settings	Configure OCSP stapling	OCSP stapling is an alternative to the Online Certificate Status Protocol (OCSP) that you can use to validate digital certificates. OCSP stapling allows Dynamic Route for CDN (DCDN) servers to retrieve OCSP details. This reduces the latency when clients send requests to validate digital certificates and minimizes the time that is consumed by clients to receive the validation responses.	Disabled
	Configure force redirect	Redirects requests from clients to L1 nodes as HTTP or HTTPS requests if HTTPS secure acceleration is enabled.	Disabled
	Configure TLS version control	Enables a Transport Layer Security (TLS) protocol version for an accelerated domain to enable the TLS handshake. Only TLS 1.0, TLS 1.1, TLS 1.2, and TLS 1.3 are supported.	Disabled
	Configure HSTS	Configures HTTP Strict Transport Security (HSTS) to force clients such as browsers to use HTTPS to connect to the server.	Disabled
	Configure a referer whitelist or blacklist to enable hotlink protection	Allows you to configure a referer blacklist or whitelist to authenticate and authorize visitors.	Disabled

Feature	Reference	Description	Default
Access control	Configure URL authentication	Allows you to configure URL authentication to prevent unauthorized downloads and access to the resources on the origin server.	Disabled
	Configure an IP address blacklist or whitelist	Allows you to configure an IP blacklist or whitelist to authenticate and authorize visitors.	Disabled
	Configure a User-Agent blacklist or whitelist	Allows you to configure a User-Agent blacklist or whitelist to authenticate and authorize visitors.	Disabled
Performance optimization	Configure HTML optimization	Compresses and removes HTML redundant content, such as blank lines and carriage return characters, to reduce the file size.	Disabled
	Configure intelligent compression	Supports smart compression for content in multiple formats to reduce the size of transmitted content.	Disabled
	Configure the parameter filtering feature	Determines whether DCDN ignores the parameters following a question mark (`?`) in the URL of a request when DCDN retrieves and caches the requested content from the origin.	Disabled
	Configure video seeking	Allows you to seek a specified position when you play video and audio content without compromising the playback quality.	Disabled

Feature	Reference	Description	Default
Advanced Settings	Configure IPv6	After you enable IPv6 in the console, IPv6 clients can send IPv6 requests to DCDN. DCDN can include the IPv6 information of the clients in back-to-origin requests.	Disabled
Websocket	Configure WebSocket	Enables WebSocket to reduce server resource and bandwidth usage and facilitate real-time communication.	Disabled

2.Copy configurations to domain names

Alibaba Cloud Dynamic Route for CDN (DCDN) allows you to copy configurations from a domain name to one or more domain names. This topic describes how to copy configurations to one or more domain names.

Scenarios

If you want to apply the same configurations to multiple domain names, you can copy the configurations from a source domain name to specified domain names. This facilitates domain name configurations.

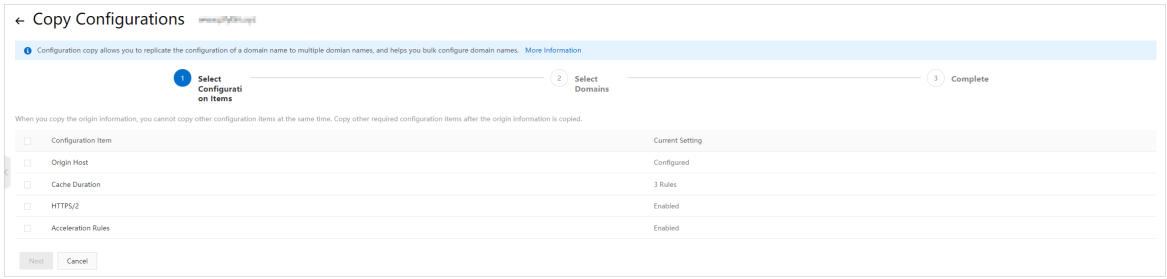
Precautions

- After you copy the configurations of one domain name to another domain name, the copy operation cannot be undone. Make sure that the configurations that you want to copy are correct.
- For domain names that have high traffic or bandwidth usage, proceed with caution to prevent unexpected financial losses.
- Special configurations that are applied through tickets cannot be copied.

Procedure

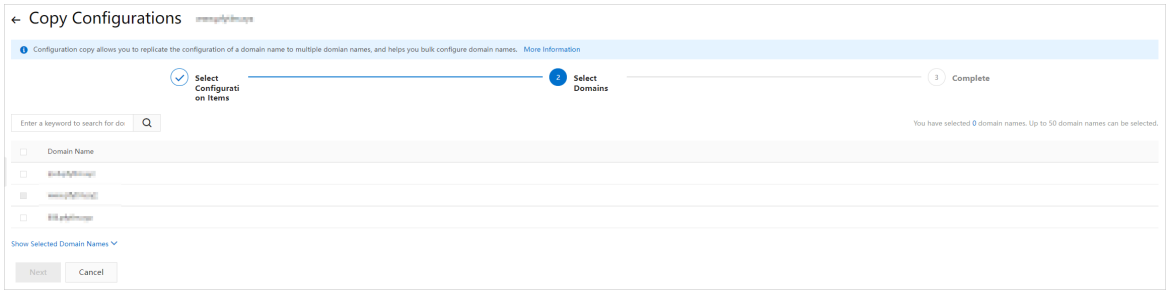
1.
2. In the left-side navigation pane, click **Domain Names**.
3. On the **Domain Names** page, find the domain name that you want to manage and click **Copy Configurations**.
4. Select the configuration items that you want to copy and click **Next**.

> Note
>
> - You cannot copy the origin information and basic information including the CNAME, business type, and acceleration region at the same time.
> - You cannot copy SSL certificates from one domain name to another domain name.
> - When you copy custom back-to-origin HTTP headers to another domain name, the HTTP headers overwrite the existing HTTP headers of the destination domain name. For example, if you set cache_control to private for Domain Name A and set cache_control to public for Domain Name B. After you copy cache_control from Domain Name B to Domain Name A, cache_control of Domain Name A is set to public.
> - When you copy configurations of feature switches, referer whitelist, referer blacklist, IP whitelist, or IP blacklist to another domain name, the copied configurations overwrite the existing configurations.



5. Select the domain names to which you want to apply the copied configurations and click **Next**.

   You can enter a keyword to search for domain names.



6. In the **Copy Configurations** dialog box, click **OK**.

3.Verify the ownership of a domain name

The first time a domain name is added to Dynamic Route for CDN (DCDN), DCDN verifies the ownership of the domain name. This ensures that the domain name is added by the owner. If a domain name that belongs to User A is added to DCDN by User B, security issues may arise. If the domain name has passed the verification, the ownership verification is not required when you add the domain name to DCDN for a second time or add the subdomains of the domain name.

Verification methods

Log on to the DCDN console and go to the **Add Domain Name** page. Set the parameters and click **Next**. You are redirected to the verification page, as shown in the following figure. You can prove the ownership by adding a DNS record or uploading a verification file. Choose one of the methods based on your business requirements. You can add a domain name to DCDN only after the domain name passes ownership verification.

Method 1: Add a DNS record to prove the ownership (recommended)

The domain name `image.example.com` is used as example to demonstrate how to verify the ownership of a domain name by adding a DNS record.

1. On the verification page, click the **Method 1: DNS Settings** tab.

   The system automatically recognizes the record type, host, and record value. Do not close the verification page before the verification process is completed.

   > **Note** If you want to add the domain name to DCDN by calling the AddDcdnDomain or BatchAddDcdnDomain operation, you must first call the DescribeDcdnVerifyContent operation to query the record value and add a TXT record for the domain name. Then, you can call the AddDcdnDomain or BatchAddDcdnDomain operation to add the domain name to DCDN.

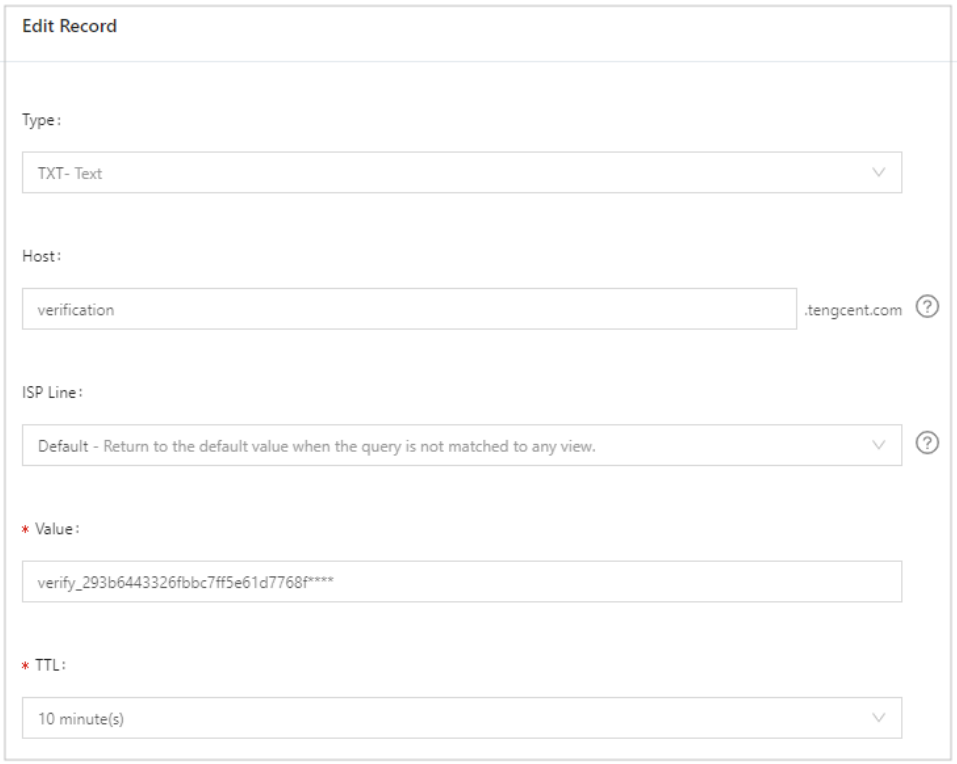
2. Add a TXT record in the system of your DNS service provider.

   Alibaba Cloud DNS is used in this topic to demonstrate how to add a TXT record. You can use similar methods to add TXT records in the systems of other DNS service providers, such as Tencent Cloud and Xinnet.

   i. Log on to the Alibaba Cloud DNS console.

   ii. Navigate to the **Manage DNS** page, find the root domain name `example.com`, and then click **Configure** in the Actions column.

iii. Click **Add Record** and enter the record type, host, and record value obtained in Step 1.



Parameter	Description	Example
Type	Select **TXT**.	TXT
Host	Enter the prefix of the domain name.	verification
ISP Line	Select the Internet service provider (ISP) of the domain name.	We recommend that you keep the default setting.
Value	Enter the record value obtained in Step 1.	verify_293b6443326fbbc7ff5e61d7768f****
TTL	Enter a time-to-live (TTL) value for the TXT record. A smaller value indicates a shorter period of time to apply record updates. The default TTL value is 10 minutes.	We recommend that you keep the default setting.

iv. Click **OK**.

3. After the TXT record takes effect, log on to the DCDN console. Click **Verify** to complete the verification process.

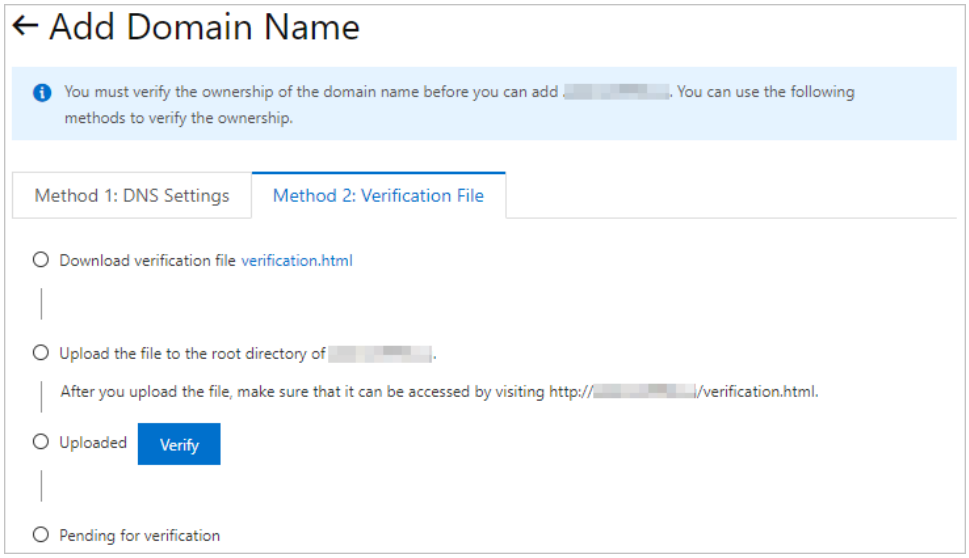
   If the system prompts that the domain name fails the verification, check whether the TXT record is correct. Wait for the TXT record to take effect and try again.

Method 2: Upload a verification file to prove the ownership

The domain name `image.example.com` is used as example to demonstrate how to verify the ownership of a domain name by adding a DNS record.

1. On the verification page, click the **Method 2: Verification File** tab.

   Do not close the verification page before the verification process is completed.



2. Click `verification.html` to download the verification file of the domain name.

   > **Note** You can call the DescribeDcdnVerifyContent operation to generate strings that must be used in the verification file. If you want to call the AddDcdnDomain or BatchAddDcdnDomain operation to add the domain name to DCDN, you must first call the DescribeDcdnVerifyContent operation to generate strings. Then, create a `verification.html` file that contains the strings and upload the file to the origin server.

3. Upload the verification file to the root directory on the origin server of the domain name. The origin server can be an Elastic Compute Service (ECS) instance, an Object Storage Service (OSS) bucket, a Cloud Virtual Machine (CVM) instance, a Container-Optimized OS (COS) instance, or an Elastic Compute Cloud (EC2) instance.

   After you upload the verification file, DCDN visits the origin server at `http://example.com/verification.html` to obtain the verification file. Then, DCDN determines whether you have uploaded the verification file as required. Make sure that the verification file is accessible.

4. Click **Verify** to complete the verification.

Related API operations

- VerifyDcdnDomainOwner: Adds a DNS record for ownership verification.
- DescribeDcdnVerifyContent: Uploads a verification file for ownership verification.

FAQ

The following issues may arise the first time a domain name is added to DCDN:

- Q: Why does Alibaba Cloud CDN verify the ownership of domain names?

  A: Ownership verification ensures that domain names are added to Alibaba Cloud CDN only by their owners. If a domain name that belongs to User A is added to Alibaba Cloud CDN by User B, security issues may arise.

- Q: If I have multiple Alibaba Cloud accounts and this is the first time a domain name is added to

Alibaba Cloud, does Alibaba Cloud CDN verify the ownership of the domain name for each account?

A: Yes. Each Alibaba Cloud account is identified as an independent user. The first time a domain name is added to Alibaba Cloud CDN, Alibaba Cloud CDN verifies the ownership of the domain name for each account (user).

- Q: If a domain name passes ownership verification after I add a DNS record or upload a verification file, can I delete the record or file?

  A: Yes. The required DNS record or file is used only for ownership verification. After the domain name passes the verification, you can delete the record or file.

- Q: Do I need to prove the ownership of a domain name that is already added to DCDN?

  A: No, you do not need to prove the ownership of existing accelerated domain names. For example, you have added the domain name *.example.com to DCDN and the Canonical Name (CNAME) record that is assigned to the domain name works in an expected manner. In this case, you are considered to own the domain name example.com. If you add a subdomain name of example.com, such as **.example.com or ***.example.com, you do not need to perform the ownership verification.

- Q: Do I need to prove the ownership of a domain name if I call the AddDcdnDomain operation to add the domain name to DCDN?

  A: Yes, you need to prove the ownership of the domain name in this case. You must first add a DNS record or upload a verification file to the root directory of the origin server of the domain name that you want to add. Then, call the AddDcdnDomain operation to add the domain name to DCDN.

- Q: What can I do if I cannot prove the ownership of my domain name by adding a DNS record or uploading a verification file to the origin server?

  A: To address this issue, you can . In the ticket, state the reason why you cannot prove the ownership through the given methods, and include the information that can be used to prove your identity as the domain name owner. Alibaba Cloud will conduct manual verification.

4.Tag management

4.1. Tag overview

This topic provides an overview of domain name tags of Alibaba Cloud Dynamic Route for CDN (DCDN). Tags do not carry special definitions. Each tag is a key-value pair that is used to mark or filter domain names. You can use tags to mark, group, filter, and manage domain names.

Limits

Tags have the following limits:

- Each tag is a `key-value` pair.
- You can add up to 20 tags to each domain name.
- Among the tags that are added to a domain name, the key of each tag must be unique. If two tags have the same key but different values, the later one overwrites the earlier one. For example, if you create the `Key1:Value1` tag and then the `Key1:Value2` tag for the domain name `example.aliyundoc.com`, only the `Key1:Value2` tag is added to `example.aliyundoc.com`.
- A key cannot start with `aliyun` or `acs:`, contain `http://` or `https://`, or be left unspecified.
- A value cannot contain `http://` or `https://`, but can be left unspecified.
- A key must be 1 to 64 Unicode characters in length.
- A value must be 0 to 128 Unicode characters in length.
- Tags are case-sensitive.

Cases

Company information

A company added 100 domain names to DCDN. These domain names belong to e-commerce, gaming, and entertainment departments, and are used to provide services such as marketing, Game A, Game B, and post-production. The company has three O&M engineers: Alice, Tom, and Sam.

Create tags

To facilitate the management of these domain names, the company decides to use tags to group the domain names. The following table describes the key-value pairs that are used to create the tags.

Tag key	Tag value
Department	E-commerce, gaming, and entertainment
Service	Marketing, Game A, Game B, and post-production
Owner	Alice, Tom, and Sam

The company can add the preceding key-value pairs to the domain names. The following table describes the key-value pairs that are added to each domain name.

> **Note** The domain names that are used in this topic are for reference only.

Domain name	Value (key is Department)	Value (key is Service)	Value (key is Owner)
domain1	E-commerce	Marketing	Sam
domain2	E-commerce	Marketing	Sam
domain3	Gaming	Game A	Alice
domain3	Gaming	Game B	Alice
domain4	Gaming	Game B	Alice
domain5	Gaming	Game B	Tom
domain6	Gaming	Game B	Tom
domain7	Gaming	Game B	Tom
domain8	Entertainment	Post-production	Sam
domain9	Entertainment	Post-production	Sam
domain10	Entertainment	Post-production	Sam

Filter domain names by tag

- If you want to search for domain names that are managed by Sam, filter domain names by using the following tag: **Owner: Sam**.
- If you want to search for domain names that belong to the gaming department and are managed by Tom, filter domain names by using the following tags: **Department: Gaming** and **Owner: Tom**.

References

You can add tags to domain names, use tags to manage or filter domain names, and remove tags from domain names. For more information, see Tag management.

4.2. Tag management

Tags are used to mark domain names. You can filter domain names by tag. If a domain name no longer needs a tag, you can remove the tag from the domain name.

Limits

Each tag is a key-value pair. Tags have the following limits:

- You can add at most 20 tags to each domain name.
- The keys of tags that are added to the same domain name must be unique.

  If two tags added to the same domain name have the same key but different values, the tag added later overwrites the one added earlier. For example, if you add tags `Key1:Value1` and `Key1:Value2` to the domain name `test.example.com`, only the tag `Key1:Value2` is added to the domain name `test.example.com`.

Add tags to domain names

You can add tags to domain names to facilitate the management of domain names.

1.
2.
3. Add tags to domain names.
   - Add tags to a domain name
     a. On the **Domain Names** page, find the domain name to which you want to add tags and move your pointer over the tag icon.
     b. In the message that appears, click **Edit**.
     c. In the **Edit Tag** dialog box, specify the key and value and click **OK**.

     You can select a tag or click **Create Tag**.

Parameter	Description
Key	The tag key can be up to 64 Unicode characters in length, and cannot contain `http://` or `https://`. It cannot start with `aliyun` or `acs:`, or be an empty string.
Value	The tag value can be up to 128 Unicode characters in length, and cannot contain `http://` or `https://`. It cannot start with `aliyun` or `acs:` but can be an empty string.

   - Add tags to multiple domain names at a time
     a. On the **Domain Names** page, select the domain names to which you want to add tags and choose **Tags > Add Tag**.
     b. In the **Add Tags** dialog box, set the tag key and value, and click **OK**.

     You can select a tag or click **Create Tag**.

Parameter	Description
Key	The tag key can be up to 64 Unicode characters in length, and cannot contain `http://` or `https://`. It cannot start with `aliyun` or `acs:`, or be an empty string.
Value	The tag value can be up to 128 Unicode characters in length, and cannot contain `http://` or `https://`. It cannot start with `aliyun` or `acs:` but can be an empty string.

Use tags to manage domain names

After you add tags to domain names, you can filter domain names by tag.

1.
2.
3. On the **Domain Names** page, click the **Select Tag** tab.
4. Select one or more tags to filter domain names.

Query data of domain names by tag

After you add tags to domain names, you can use tags to filter domain names and query data such as data usage of the domain names.

1.
2. Filter domain names and query data.
   i. In the left-side navigation pane, choose **Monitoring > Monitoring**.
   ii. On the **Monitoring** page, click **Select Tag**.

   > **Note** If you select multiple tags, only the domain names that contain all the selected tags are returned by the system.

   iii. Select the tags that are used to filter domain names and click **Search**.

Remove tags from domain names

If a domain name no longer needs a tag, you can remove the tag from the domain name.

1.
2.
3. Select the domain name from which you want to remove tags and choose **Tags > Delete Tag**.
4. In the **Delete Tags** dialog box, select the tags that you want to remove and click **OK**.

   > **Note** You can remove a maximum of 20 tags from domain names at a time.

5. On the **Domain Names** page, click the refresh icon to check whether the tags are removed from the domain names.

Related API operations

The following table describes the API operations that you can perform to manage tags.

Feature	Description	API
Add tags	Adds tags to domain names to mark or group the domain names.	TagDcdnResources
Use tags to manage domain names	Filters domain names by tag. You can use tags to group and manage domain names.	DescribeDcdnTagResources
Remove tags	Removes tags from one or more domain names.	UntagDcdnResources

5.Basic Settings

5.1. Overview

Alibaba Cloud Dynamic Route for CDN (DCDN) allows you to configure accelerated domain names. You can view information about accelerated domain names and origin servers, change the accelerated region of domain names, and modify origin server information in the Alibaba Cloud DCDN console.

The following table lists operations that you can perform in the DCDN console.

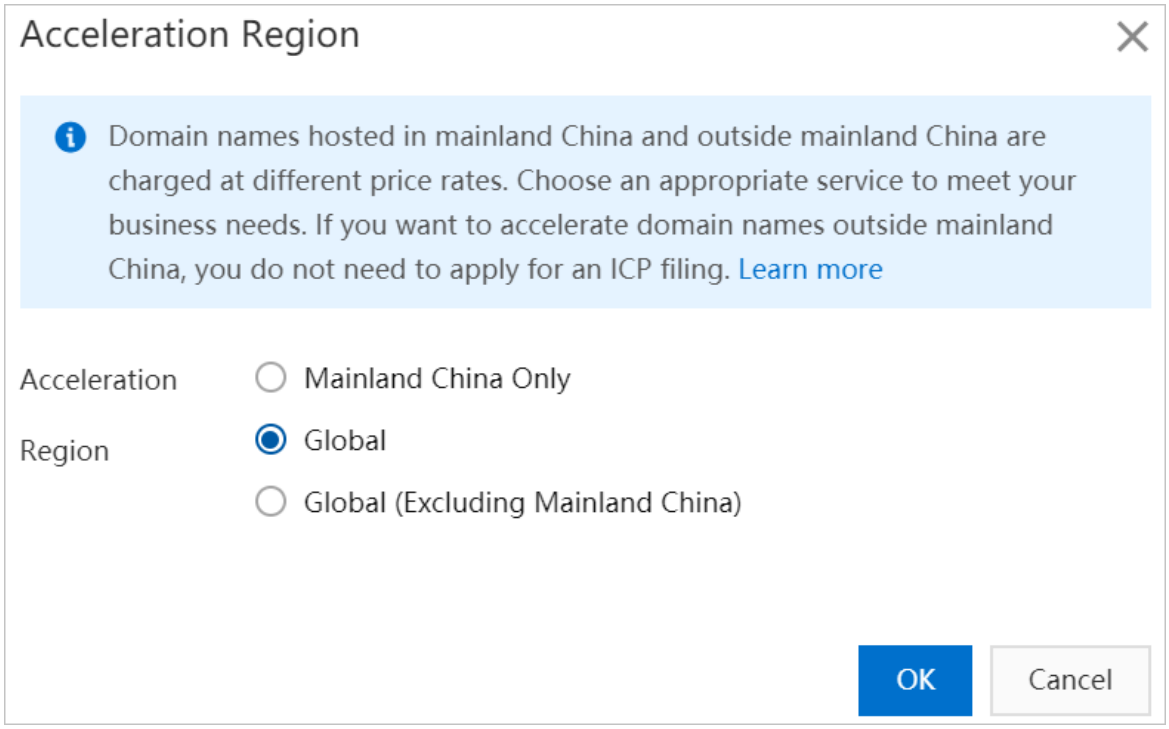
Operation	Description
Modify basic information	Changes the accelerated region.
Configure an origin server	Modifies the type, address, priority, weight, and port of an origin server.

5.2. Modify basic information

You can change the acceleration region of your Dynamic Route for CDN (DCDN) service.

Procedure

1.
2.
3. On the **Domain Names** page, find the domain name that you want to manage, and click **Configure** in the Actions column.
4. On the **Basic Settings** tab, find the **Basic Information** section.
5. In the **Basic Information** section, click **Modify** next to **Acceleration Region**.
6. In the **Acceleration Region** dialog box, select the region you want to switch to.



Parameter	Description
Mainland China Only	If you select **Mainland China Only**, you must apply for an Internet content provider (ICP) filing with the Ministry of Industry and Information Technology (MIIT) of China. For more information, see Limits.
Global	If you select **Global**, you must apply for the ICP filing with the MIIT of China. For more information, see Limits.
Global (Excluding Mainland China)	If you select **Global (Excluding Mainland China)**, no ICP filing is required.

7. Click **OK**.

5.3. Configure an origin server

Alibaba Cloud CDN supports the following types of origin server: Object Storage Service (OSS) endpoints, IP addresses of origin servers, and domain names of origin servers. You can specify one or more origin servers of each type and specify primary and secondary origin servers to balance loads. This topic describes how to add an origin server, modify the information about an origin server, and configure a health check policy for origin servers.

Usage notes

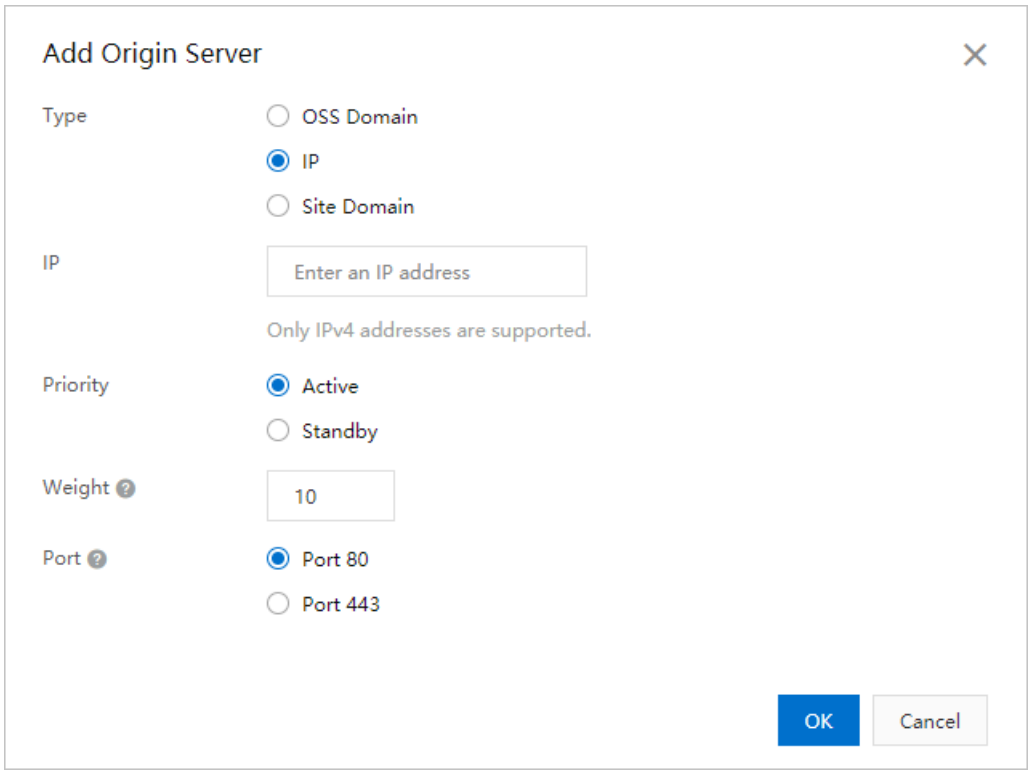
- When Alibaba Cloud DCDN retrieves resources from an origin server, the origin server is billed for data transfer. For example, if the origin server is a data center, the data center is billed for data transfer and bandwidth resources. If the origin server is an Object Storage Service (OSS) bucket, the OSS bucket is billed for data transfer.
- DCDN supports switchover between primary and secondary origin servers. If multiple origin servers are configured, DCDN preferentially redirects requests to the origin server whose **Priority** is **Primary**. If

the primary origin server fails three consecutive health checks, Alibaba Cloud CDN redirects requests to the origin server whose **Priority** is **Secondary**. If the primary origin server passes the health check, the system marks the origin server as available and restores the priority of the origin server to primary. If you set the same priority for all origin servers, DCDN automatically redirects requests to origin servers in turn.

> **Note** Layer 4 health checks are performed on origin servers. Port 80, port 443, or custom ports of origin servers are probed. Probes are sent every 2.5 seconds. If an origin server fails three consecutive probes, the system marks the origin server as unavailable.

Add an origin server or modify the information about an origin server

1.
2.
3.
4. On the **Basic Settings** tab, find the **Origin Information** section.
5. In the **Origin Information** section, click **Modify**.
6. In the **Add Origin Server** dialog box, set the following parameters.



7. Click **OK**.

6.Origin Fetch Settings

6.1. Overview

This topic provides an overview of origin fetch and its application scenarios.

What is origin fetch?

If a client requests resources that are not cached on Dynamic Route for CDN (DCDN) nodes, the request to retrieve the resources is redirected to the origin server. This process is called origin fetch. Static resources are cached on edge nodes, but dynamic resources are directly returned to the client. If you run a prefetch task on edge nodes, the edge nodes retrieve resources from the origin server and cache the resources. You can configure origin fetch based on your business requirements.

Scenarios

Origin fetch helps you manage domain names in the following scenarios.

Feature	Description	Reference
Specify a server domain name for origin fetch	If the IP address of your origin server is associated with multiple domain names or sites, you can modify the HOST header in HTTP requests to specify the site to which requests are redirected. DCDN redirects requests to the site that is specified by the HOST header.	Configure an origin host
Set the origin protocol policy	If a client requests resources that are not cached on edge nodes, DCDN redirects the request to the origin server over the protocol that is specified by the origin protocol policy. You can set the origin protocol policy to Match Client, HTTP, or HTTPS.	Configure the static origin protocol policy
Retrieve content from private Object Storage Service (OSS) buckets	If your origin server is a private Object Storage Service (OSS) bucket, you must enable the private bucket origin feature and grant DCDN permissions to access the OSS bucket. Permission control can prevent hotlinking.	Configure private bucket origin
Specify a site for origin fetch	If edge nodes access your origin server over HTTPS and the IP address of the origin server is associated with multiple domain names, you can configure Server Name Indication (SNI) to specify the domain name that edge nodes can access.	Configure an origin SNI
Validate client requests and reject requests that are not in the whitelist to access the origin server	When DCDN nodes connect to origin servers over HTTPS, the system compares common names in the certificates that are returned by the origin servers with the SNI values included in client requests. If an SNI value that is included in the client request does not match the common name in the certificate that is returned by the origin server, the client request is rejected.	Configure a common name whitelist

Feature	Description	Reference
Return partial content within the specified range from the origin server	You can enable Range Origin Fetch to accelerate the delivery of large files. This reduces the usage of back-to-origin data transfer and shortens the resource response time.	Configure range origin fetch
Specify a maximum timeout period for back-to-origin requests	You can configure the amount of time that an edge node waits for a response after a request is redirected to the origin server. The default timeout period is 30 seconds. If the edge node does not receive a response within the specified timeout period, the connection between the edge node and origin server closes.	Configure a timeout period for origin fetch requests
Add, modify, or delete an HTTP header from back-to-origin requests	You can add HTTP headers to or delete HTTP headers from back-to-origin HTTP requests.	Customize an origin HTTP header

6.2. Configure an origin host

If multiple sites are hosted on your origin server, and the site for which back-to-origin routing is enabled is different from the one to which the accelerated domain name points, you must configure an origin host. An origin host specifies the site to which Alibaba Cloud Dynamic Route for CDN (DCDN) redirects requests.

Context

Differences between an origin server and an origin host:

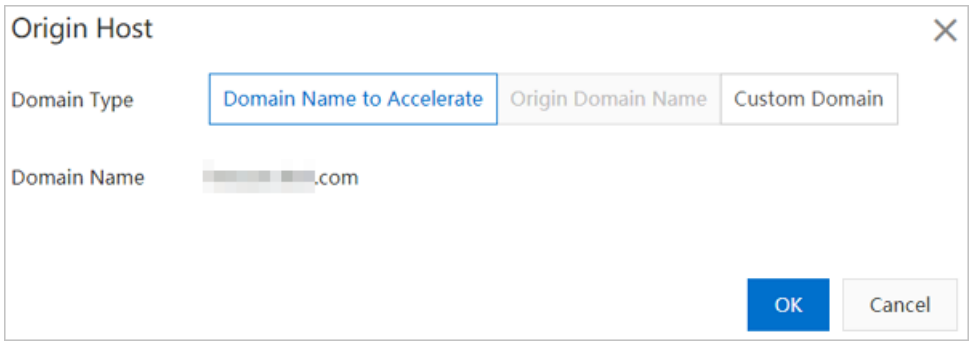
- An origin server is where your workloads run. If you set the origin server as the destination for back-to-origin requests, the requests are redirected to the IP address of the origin server.
- An origin host is the value of the HOST header in back-to-origin requests. If you set the origin host as the destination for back-to-origin requests, the requests are redirected to the site defined by the origin host.

> **Notice**
>
> - For a specific domain name, the origin host is the accelerated domain name by default.
> - For a wildcard domain name, the origin host is the domain name that matches the wildcard domain and is actually visited by users. For example, if the wildcard domain name is *.aliyundoc.com and the domain name that is visited by users is example.aliyundoc.com, the origin host is example.aliyundoc.com.

Procedure

1.
2.
3. On the **Domain Names** page, find the domain name that you want to manage and click **Configure** in the Actions column.

4.
5. On the **Origin Fetch** tab, find **Origin Host**.
6. Turn on **Origin Host** and specify **Domain Type**.



Parameter	Description
Domain Name to Accelerate	The domain name that users visit.
Origin Domain Name	The domain name of the origin server. **Note** - If you set the type of the origin server to IP when you add an origin server, **Origin Domain Name** is dimmed. - If you set the type of the origin server to OSS Domain when you add an origin server, **Origin Host** is turned on and Domain Type is set to **Origin Domain Name**.
Custom Domain	You can specify a domain name. **Note** - Make sure that the custom domain name is associated with the origin server. Otherwise, back-to-origin routing fails. - If your origin server is associated with multiple domain names, you must specify a domain name to which requests are redirected. Otherwise, back-to-origin routing fails.

7. Click **OK**.

Configuration examples

Example 1: The address of the origin server is a domain name.

Domain name	Description
Accelerated domain name: `image.example.com` Address of the origin server: `example.com`	• If you set Domain Type to **Domain Name to Accelerate**, back-to-origin requests are redirected to `image.example.com` that is hosted on the origin server `example.com`. • If you set Domain Type to **Origin Domain Name**, back-to-origin requests are redirected to the origin server `example.com`. • If you set Domain Type to **Custom Domain**, the origin host is the custom domain name that you specify.

Example 2: The address of the origin server is an IP address.

Domain name	Description
Accelerated domain name: `example.com` Address of the origin server: `10.10.10.10`	• If you set Domain Type to **Domain Name to Accelerate**, back-to-origin requests are redirected to `example.com` that is hosted on the origin server `10.10.10.10`. • If you set Domain Type to **Custom Domain**, back-to-origin requests are redirected to the domain name that you specify.

Example 3: The address of the origin server is an OSS domain name.

Domain name	Description
Accelerated domain name: `example.com` Address of the origin server: `example.oss-cn-hangzhou.aliyuncs.com`	• If you set Domain Type to **Domain Name to Accelerate**, back-to-origin requests are redirected to `example.com` on the origin server `example.oss-cn-hangzhou.aliyuncs.com`. • If you set Domain Type to **Origin Domain Name**, back-to-origin requests are redirected to the origin server `example.oss-cn-hangzhou.aliyuncs.com`. • If you set Domain Type to **Custom Domain**, back-to-origin requests are redirected to the domain name that you specify.

Related information

- BatchSetDcdnDomainConfigs

6.3. Configure the static origin protocol policy

Alibaba Cloud Dynamic Route for CDN (DCDN) allows you to configure the protocol that is used by DCDN nodes to retrieve static resources from origin servers. After you configure the static origin protocol policy, DCDN redirects HTTP requests to the origin server over port 80 or HTTPS requests to the origin server over port 443 based on the protocol that you specify.

Context

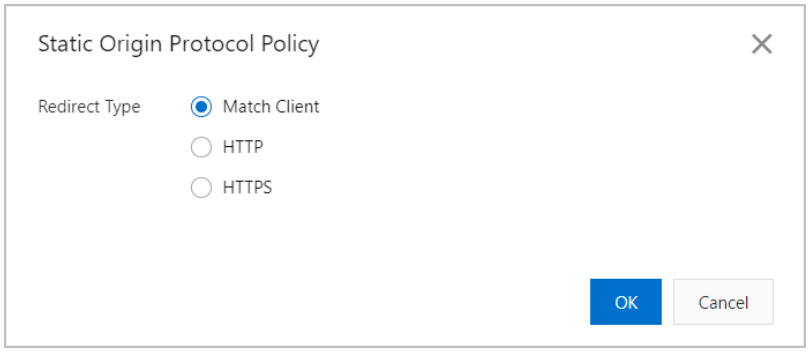
By default, the static origin protocol policy is disabled. In this case, requests are redirected to the origin server over the port that is specified in **Basic Settings > Origin Information**.

- If port 443 is used, requests are redirected to the origin server over HTTPS.
- If port 80 or a custom port is used, requests are redirected to the origin server over HTTP.

> **Note** The static origin protocol policy supports only port 80 and port 443. After the static origin protocol policy is enabled, the port that is specified in **Origin Information** becomes invalid. If you want DCDN nodes to redirect requests to the origin server over a custom port, submit a ticket.

Procedure

1.
2.
3.
4.
5. On the **Origin Fetch** tab, find **Static Origin Protocol Policy**.
6. Turn on **Static Origin Protocol Policy**.
7. In the **Static Origin Protocol Policy** dialog box, set Redirect Type to **Match Client**, **HTTP**, or **HTTPS**.



Parameter	Description
Match Client	When a client sends an HTTP or an HTTPS request, DCDN redirects the request to the origin server over the protocol used by the client. Make sure that both port 443 and port 80 are open on the origin server. Otherwise, DCDN may fail to redirect client requests to the origin server. **Note** The static origin protocol policy protects data from tampering and leakage during transmission. If you want to transmit sensitive data such as user identity data over HTTPS, and other data such as image files over HTTP, we recommend that you set Redirect Type to **Follow**.
HTTP	DCDN redirects requests to the origin server over HTTP.

Parameter	Description
HTTPS	DCDN redirects requests to the origin server over HTTPS. Note ◦ HTTPS encryption consumes additional computing resources on the origin server. ◦ If you set Redirect Type to HTTPS, port 443 is used by default. If you want to use a custom port, submit a ticket.

8. Click OK.

Related information

- BatchSetDcdnDomainConfigs

6.4. Configure private bucket origin

If your origin server is a private Object Storage Service (OSS) bucket, you must enable the private bucket origin feature and grant Dynamic Route for CDN (DCDN) permissions to access the OSS bucket. This feature can be used for access authentication and protect origin servers from unauthorized access. This topic describes how to enable and disable access to private OSS buckets.

Background information

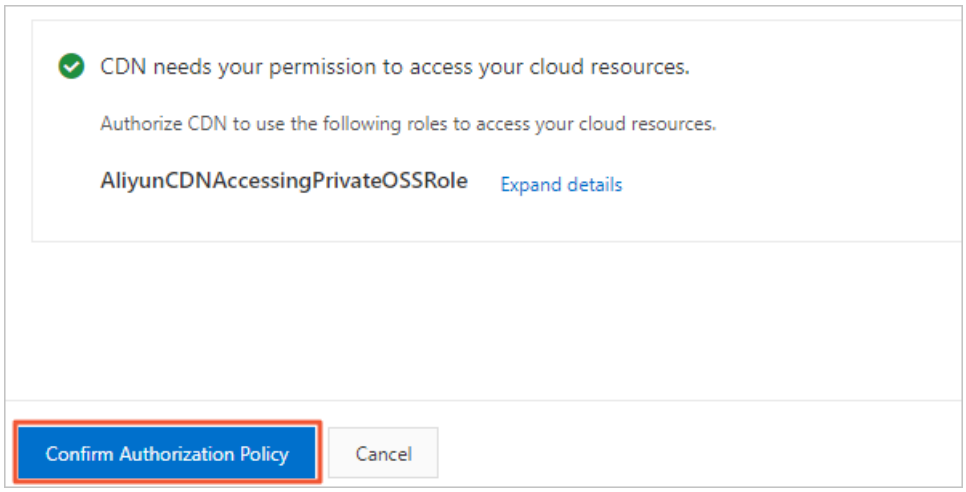
After you grant DCDN permissions to access private OSS buckets, you can also use features such as hotlink protection and URL authentication provided by DCDN to protect resources from unauthorized access. For more information, see Configure a referer whitelist or blacklist to enable hotlink protection and Configure URL authentication.

Notice

- After you grant DCDN permissions to access private OSS buckets, DCDN is granted read-only permissions on all your OSS buckets.
- After you enable the private bucket origin feature and grant DCDN permissions to access private OSS buckets, DCDN can access all resources in your private OSS buckets by using the accelerated domain names. Proceed with caution when you use this feature. Do not enable the private bucket origin feature or grant DCDN permissions to access private OSS buckets if your private bucket is unsuitable as an origin for your domain name.
- If your website is vulnerable to attacks, we recommend that you purchase the Anti-DDoS service. Do not enable the private bucket origin feature or grant DCDN permissions to access private OSS buckets.
- The private bucket origin feature conflicts with the settings of the default homepage of the static website that is hosted on OSS. For more information about how to use the private bucket origin feature and the static website hosting feature at the same time, see Why do requests destined for my accelerated domain name trigger the error message "You are forbidden to list buckets" after access to private Object Storage Service (OSS) is enabled?

Enable access to private OSS buckets

1.
2.
3.
4.
5. (Optional) The first time you grant DCDN permissions to access private OSS buckets, this step is required. In the **Alibaba Cloud OSS Private Bucket Access** section, click **Authorize**, and then click **Confirm Authorization Policy**.



6. In the **Alibaba Cloud OSS Private Bucket Access** section, turn on **Alibaba Cloud OSS Private Bucket Access**.

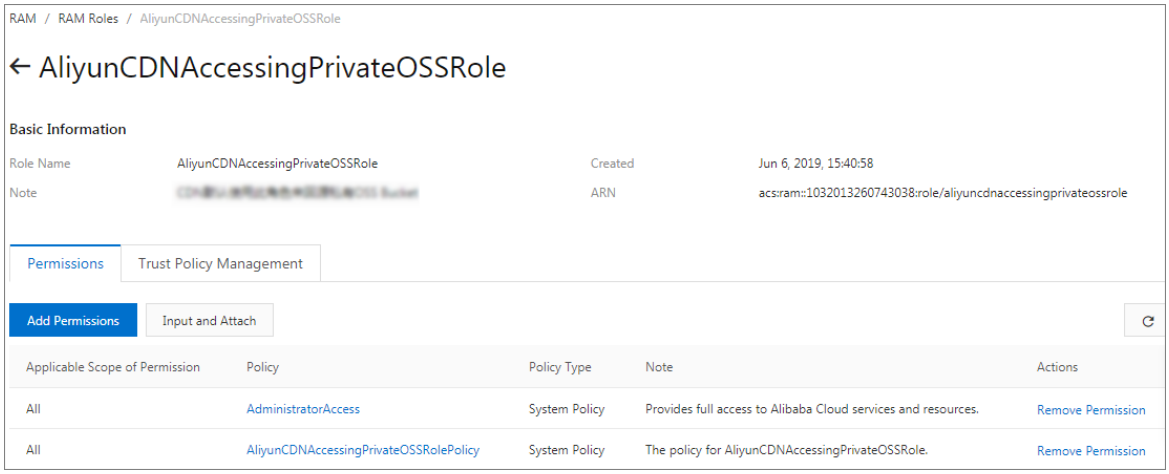
> **Note** You need only to perform the preceding steps if you want to authorize DCDN to access unencrypted files in a private OSS bucket. If you want DCDN to access OSS objects that are encrypted by using Key Management Service (KMS), you must first attach the **AliyunKMSCryptoUserAccess** policy to the RAM role **AliyunCDNAccessingPrivateOSSRole**.

7. (Optional) Attach the **AliyunKMSCryptoUserAccess** policy to the RAM role **AliyunCDNAccessingPrivateOSSRole**.
   i. Log on to the RAM console.
   ii. In the left-side navigation pane, choose **Identities > Roles**.
   iii. In the **Role Name** column, find the RAM role **AliyunCDNAccessingPrivateOSSRole**.
   iv. Click **Add Permissions** in the Actions column. In the Add Permissions panel, the value of the **Principal** field is automatically specified.
   v. Click **System Policy** and enter **AliyunKMSCryptoUserAccess** in the search box to search for the **AliyunKMSCryptoUserAccess** permission policy. Click the permission policy to add it to the **Selected** list.
   vi. Click **OK**.
   vii. Click **Complete**.

Disable access to private OSS buckets

If you no longer need an accelerated domain name to access your private OSS buckets, you can log on to the RAM console and revoke the access permissions that are granted to DCDN.

1. Log on to the RAM console.
2. In the left-side navigation pane, choose **Identities > Roles**.
3. In the **Role Name** column, find the RAM role **AliyunCDNAccessingPrivateOSSRole**.



4. Revoke all permissions that are granted to the RAM role **AliyunCDNAccessingPrivateOSSRole**.
   i. Click **Remove Permission** in the Actions column.
   ii. In the Remove Permission message, click **OK**.
5. Choose **Identities > Roles** and delete **AliyunCDNAccessingPrivateOSSRole**.
   i. Find the RAM role **AliyunCDNAccessingPrivateOSSRole** and click **Delete** in the Actions column.

ii. In the **Delete Role** message, click **OK**.

6.5. Configure an origin SNI

If your origin IP address is bound to multiple domains, you must set a Server Name Indication (SNI) value to ensure that the Dynamic Route for CDN (DCDN) node can access your origin server over HTTPS.

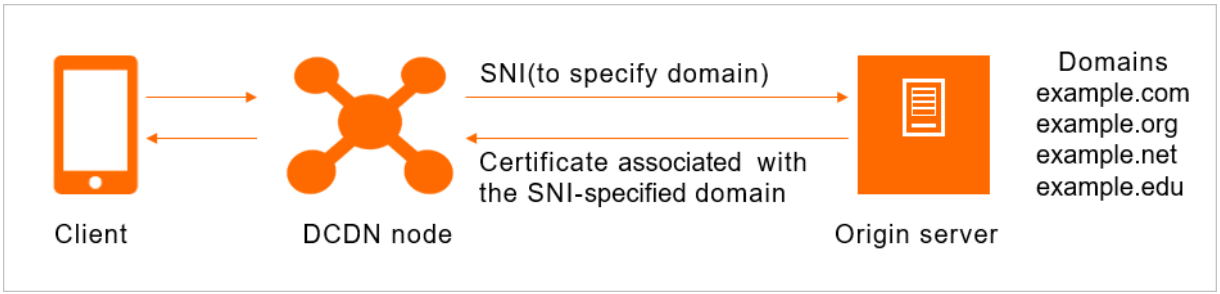
Context

SNI is an extension of Transport Layer Security (TLS) by which a client determines which hostname it is attempting to connect to at the beginning of the handshake process. This allows a server to present multiple certificates on the same IP address and TCP port. In this way, multiple HTTPS websites (or any other service over TLS) that have different certificates can be served by the same IP address.

If your origin server uses one IP address to provide HTTPS service for multiple domains and you have specified port 443 for DCDN to communicate with the origin server, you must set an SNI value to specify the requested domain. This way, when a DCDN node accesses your origin server over HTTPS, the server returns the correct certificate of the requested domain.

> **Note** If your origin is an Alibaba Cloud Object Storage Service (OSS) bucket, you do not need to set an SNI value.

The following figure shows how SNI works.

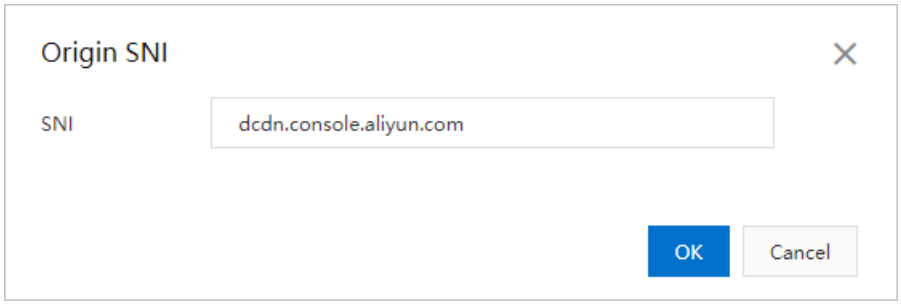


1. The DCDN node sends an HTTPS access request to the origin server. The requested domain is included in SNI.
2. After the origin server receives the request, it returns the certificate of the requested domain to the DCDN node.
3. After the DCDN node receives the certificate, it establishes a secure connection to the origin server.

Procedure

1.
2.
3.
4.
5. On the **Origin Fetch** tab, find **Origin SNI**.
6. Turn on **Origin SNI**, and enter the name of the domain to be requested.

In Alibaba Cloud DCDN, SNI specifies a domain name of your origin server. If your origin server uses one IP address to provide HTTPS services for multiple domains, you must set an SNI value to specify the requested domain name, for example, `cdn.console.aliyun.com`.



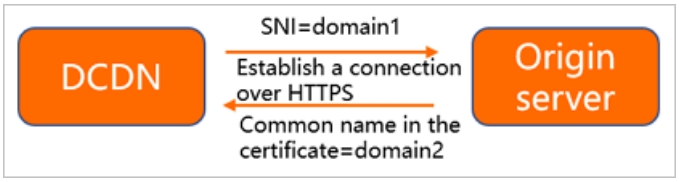
7. Click **OK**.

6.6. Configure a common name whitelist

When Dynamic Route for CDN (DCDN) nodes connect to origin servers over HTTPS, the system compares common names in the certificates returned by the origin servers with Server Name Indication (SNI) values included in client requests. To pass the origin certificate verification and connect the DCDN nodes to the origin servers, you can enable and configure the common name whitelist feature.

Context

A common name refers to the specific website domain name that is used to apply for a Secure Sockets Layer (SSL) certificate. The following figure shows that a client request is rejected when the SNI value included in the request does not match the common name in the certificate returned by the origin server. As a result, the DCDN node fails to connect to the origin server over HTTPS. However, if you enable the common name whitelist feature and add domain2 to the common name whitelist, the DCDN node can connect to the origin server over HTTPS.



Procedure

1.
2.
3.
4.
5. On the **Origin Fetch** tab, find **Common Name Whitelist--Beta**, and turn on the **Status** switch.
6. Enter the domain name that you want to add to the common name whitelist.

> **Note** You can enter multiple domain names and separate them with commas (,). For example, you can enter `example.com,example.org,example.net`.

7. Click **OK**.

6.7. Configure range origin fetch

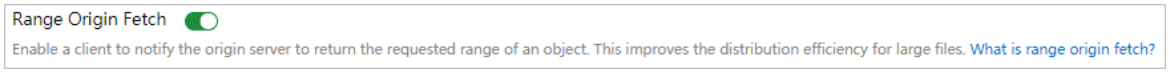
The range origin fetch feature enables Dynamic Route for CDN (DCDN) to retrieve resources from the origin server based on HTTP range requests. The origin server then returns partial content within the specified range. This accelerates delivery of large files. This feature reduces the back-to-origin data usage and shortens the resource response time. This topic describes how to configure range origin fetch.

Context

Make sure that the origin server supports HTTP range requests. If the HTTP request header contains the range field, the origin server can return 206 Partial Content.

Procedure

1.
2.
3.
4.
5. On the **Origin Fetch** tab, find **Range Origin Fetch**.
6. Turn on or off **Range Origin Fetch**.



Range Origin Fetch	Description	Example
On	If you need to access the specified part of a resource file, turn on **Range Origin Fetch** to improve resource response efficiency. After **Range Origin Fetch** is turned on, when DCDN receives a byte-range request and cannot return the requested resources from the cache, DCDN forwards the request that contains the **range** field to the origin server. When the origin server receives the request, the origin server returns a file that has the specified number of bytes based on the range field. Then, DCDN returns the file to the client.	If a client sends a request that contains `range:0-100` to DCDN, the request that is received by the origin server from DCDN contains `range:0-100`. Based on the range field, the origin server returns a file that has 101 bytes in the range of 0 to 100 to DCDN. Then, DCDN returns the file to the client.

Range Origin Fetch	Description	Example
Off	If you need to access all the content of the resource file, turn off **Range Origin Fetch**. After **Range Origin Fetch** is turned off, when DCDN receives a byte-range request and cannot return the requested resources from the cache, DCDN forwards the request that does not contain the range field to the origin server. When the origin server receives the request, the origin server returns the entire file to DCDN. Then, DCDN returns the requested range to the client. After the client receives the requested range, the client automatically closes the HTTP connection to DCDN. As a result, the file returned from the origin server is not cached on DCDN. This decreases the cache hit ratio and increases the back-to-origin data usage.	If a client sends a request that contains `range: 0-100` to DCDN, the request that is received by the origin server from DCDN does not contain the range field. Then, the origin server returns the entire file to DCDN, and DCDN returns 101 bytes to the client based on the range field. When the client receives the requested range, the client is disconnected from DCDN. As a result, the file that is returned from the origin server is not cached on DCDN.

6.8. Configure a timeout period for origin fetch requests

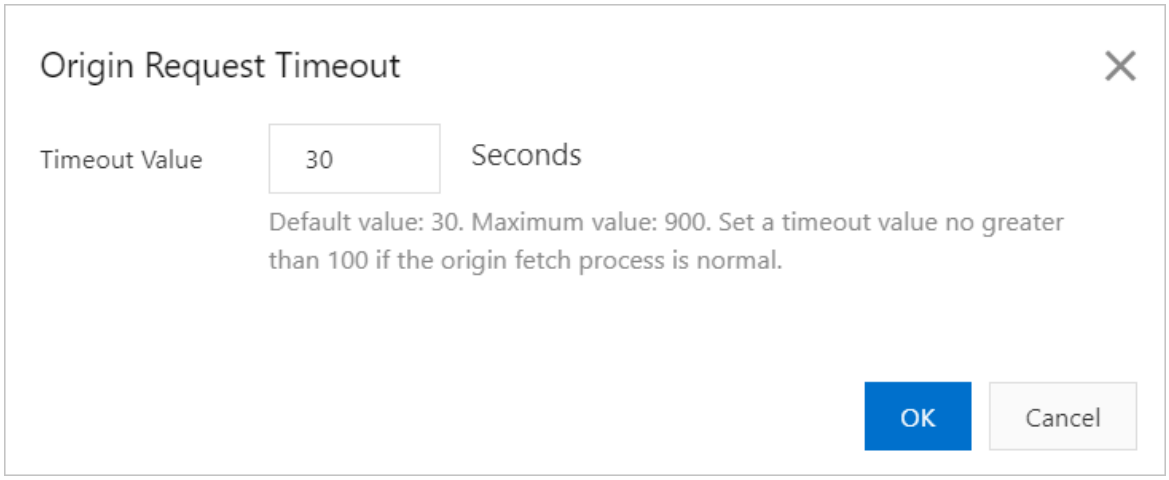
When Alibaba Cloud Dynamic Route for CDN (DCDN) requests resources from an origin server, the default timeout period is 30 seconds. If an origin fetch request times out, the request fails. You can configure a custom timeout period based on your business requirements to ensure that origin fetch requests can work as expected. This topic describes how to configure a timeout period for origin fetch requests.

Precautions

The default request timeout period between L1 nodes (edge nodes) and L2 nodes (aggregate nodes) is 36 seconds. Therefore, the default timeout period for requests that are sent from L1 nodes to L2 nodes and then reach the origin servers is 36 seconds. If you want to set a longer timeout period for back-to-origin routing, you must submit a ticket.

Procedure

1.
2.
3.
4.
5. On the **Origin Fetch** tab, find **Origin Request Timeout** and click **Modify**.
6. In the **Origin Request Timeout** dialog box, configure **Timeout Value**.



7.

Related information

- BatchSetDcdnDomainConfigs

6.9. Customize an origin HTTP header

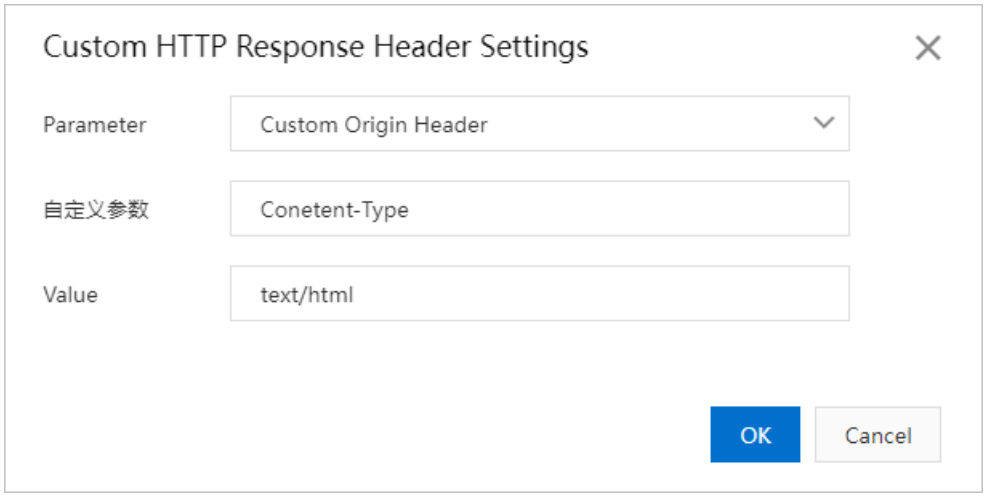
HTTP headers are the components of the header section of request and response messages that are transmitted over HTTP. HTTP headers define the resources being requested, the behavior of the client or server, and the parameters of an HTTP transaction. You can add or remove HTTP headers if you configure DCDN to communicate with the origin over HTTP.

Context

HTTP headers include general headers, request headers, and response headers.

Procedure

1.
2.
3.
4.
5. Click the **Custom Origin HTTP Header** tab.
6. Click **Add**.
7. In the **Custom HTTP Response Header Settings** dialog box, select a parameter from the **Parameter** drop-down list, and enter a value in the **Value** field.



8. Click **OK**.

6.10. Rewrite URIs in back-to-origin requests

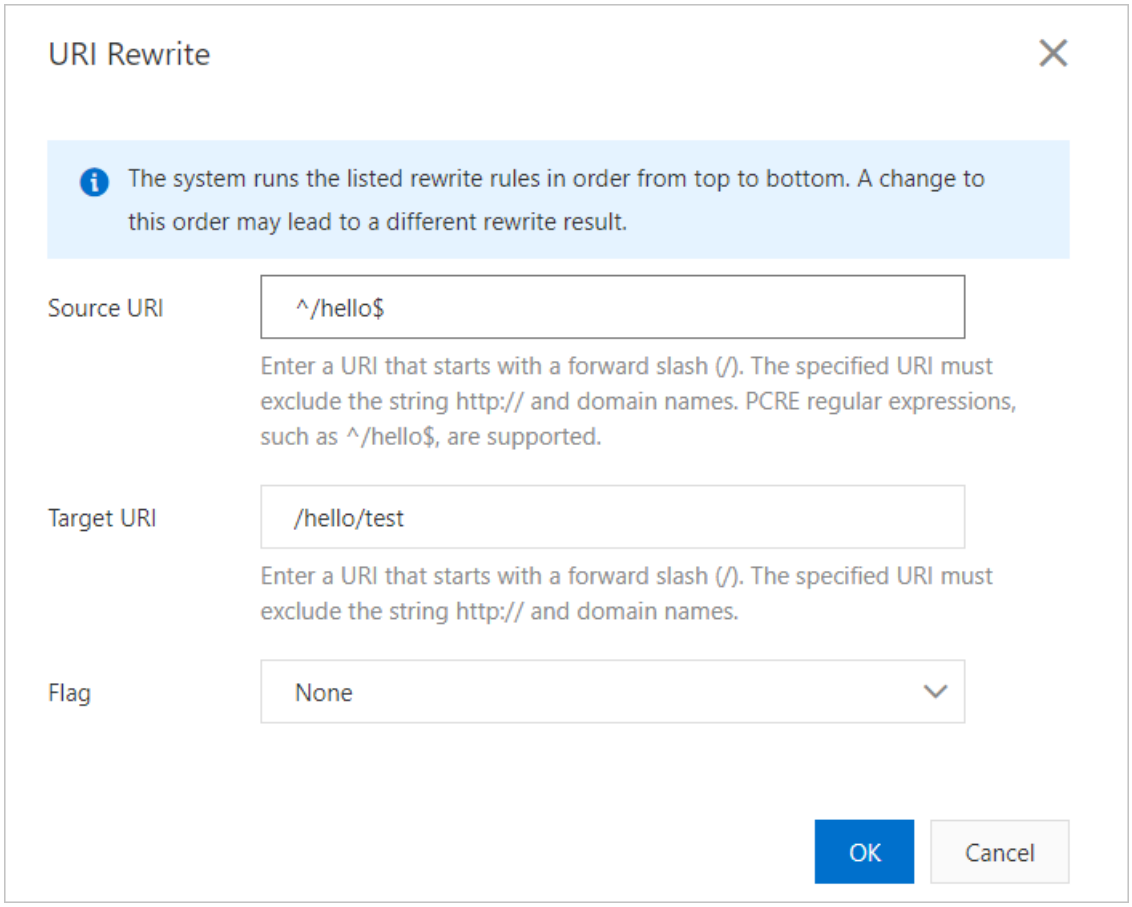
If you want to rewrite the Uniform Resource Identifier (URI) in back-to-origin requests, you can create rules to rewrite URIs. This topic describes how to configure rules to rewrite URIs in the Alibaba Cloud Dynamic Route for CDN (DCDN) console.

Context

If a request URI does not match the URI of the requested resource on an origin server, you must rewrite the request URI. You can create multiple rewrite rules based on your business requirements.

Procedure

1.
2.
3.
4.
5. Click the **URI Rewrite** tab.
6. On the **URI Rewrite** tab, click **Add**.
7. In the URI Rewrite dialog box that appears, specify the source URI, the target URI, and the flag.



Parameter	Example	Description
Source URI	^/hello$	Enter a URI that starts with a forward slash (/). The URI cannot contain http:// or domain names. Perl Compatible Regular Expressions (PCRE) are supported.
Target URI	/hello/test	Enter a URI that starts with a forward slash (/). The URI cannot contain http:// or domain names.
Flag	None	If multiple rules are configured, the system continues to match the request against the subsequent rules even after the current rule is matched.
	break	If this rule is matched, only the URI is rewritten. Then, the system stops matching rules.
	enhance_break	If this rule is matched, the URI and its parameters are rewritten. Then, the system stops matching rules.

> **Notice**
>
> - If you set the flag of a **URI Rewrite** rule to break, the query parameters in the request URI is not rewritten. However, the settings of **Parameter Rewrite** still take effect.
> - If you set the flag of a **URI Rewrite** rule to enhance_break, the parameter rewrite settings may conflict with the settings of the **Parameter Rewrite** feature. If you configure both features for the same domain name, make sure that the settings do not conflict with each other.
> - If you set the flag of a **URI Rewrite** rule to enhance_break, the parameter rewrite settings may conflict with the settings of **Retain Parameters** or **Ignore Parameters** on the **Domain Names > Optimization** page. If you configure these three features at the same time, make sure that the settings do not conflict with each other.

8. Click **OK** to apply and run the rewrite rule.

   To modify or delete a rewrite rule, find the rule on the **URI Rewrite** tab and click **Modify** or **Delete** in the Actions column.

> **Notice**
>
> - You can configure up to 50 **URI Rewrite** rules for a domain name.
> - The system runs the rewrite rules that are listed on the URI Rewrite tab in order from top to bottom. If you change the order of the rewrite rules, you may obtain a different result.
> - The **URI Rewrite** feature is different from the **Rewrite** feature on the Caching page. The **Rewrite** feature is performed on the edge nodes. This feature affects the internal links of Alibaba Cloud CDN, and rewrites cache keys. The **URI Rewrite** feature is performed on the origin nodes. This feature does not affect the internal links of Alibaba Cloud CDN or rewrite the cache keys.

Example 1

Source URI	^/hello$
Target URI	/index.html
Flag	None
Description	Original request: `http://aliyundoc.com/hello` Final request: `http://aliyundoc.com/index.html` The system continues to match this request against the subsequent URI rewrite rules that are listed on the **URI Rewrite** tab.

Example 2

Source URI	^/hello.jpg$
Target URI	/image/hello.jpg

Flag	break
Description	Original request: `http://aliyundoc.com/hello.jpg` Final request: `http://aliyundoc.com/image/hello.jpg` The system stops matching this request against the subsequent URI rewrite rules that are listed on the **URI Rewrite** tab.

Example 3

Source URI	^/hello.jpg?code=123$
Target URI	/image/hello.jpg?code=321
Flag	enhance_break
Description	Original request: `http://aliyundoc.com/hello.jpg?code=123` Final request: `http://aliyundoc.com/image/hello.jpg?code=321` The system stops matching this request against the subsequent URI rewrite rules that are listed on the **URI Rewrite** tab.

7.Dynamic and static acceleration rules

7.1. Overview

If dynamic acceleration is enabled, you can create custom acceleration rules for static and dynamic content. In this case, static content is cached on edge nodes and requests for dynamic content are redirected to the origin server over the optimal route. If dynamic acceleration is disabled, the acceleration for dynamic content delivery becomes unavailable, but static content is still cached on edge nodes.

The following table lists the operations that you can perform to configure acceleration rules for static and dynamic content delivery.

Operation	Description
Configure static file types	Allows you to configure acceleration rules for static content by specifying the types of files that can be cached on edge nodes. This prevents static content from occupying resources that are reserved for the acceleration of dynamic content delivery.
Specify static file URIs	Allows you to configure acceleration rules for static content by specifying the Uniform Resource Identifiers (URIs) of files that can be cached on edge nodes. This prevents static content from occupying resources that are reserved for the acceleration of dynamic content delivery.
Specify static file paths	Allows you to configure acceleration rules for static content by specifying the paths of files that can be cached on edge nodes. This prevents static content from occupying resources that are reserved for the acceleration of dynamic content delivery.
Configure back-to-origin requests to retrieve dynamic content	Allows you to configure dynamic content retrieval settings, including the protocol used by the client to retrieve dynamic content from the origin server and whether to enable load balancing. You can configure the settings based on your business requirements.

7.2. Configure static file types

To accelerate the delivery of both static and dynamic resources, you can enable the dynamic acceleration feature. You can customize acceleration rules for static resources by configuring static file types. In this case, static files no longer use dynamic acceleration. This allows you to cache static resources on edge nodes and to retrieve dynamic resources from the origin server over an optimal route.

Context

The following acceleration rules are applicable to both static and dynamic resources:

- Enable dynamic acceleration

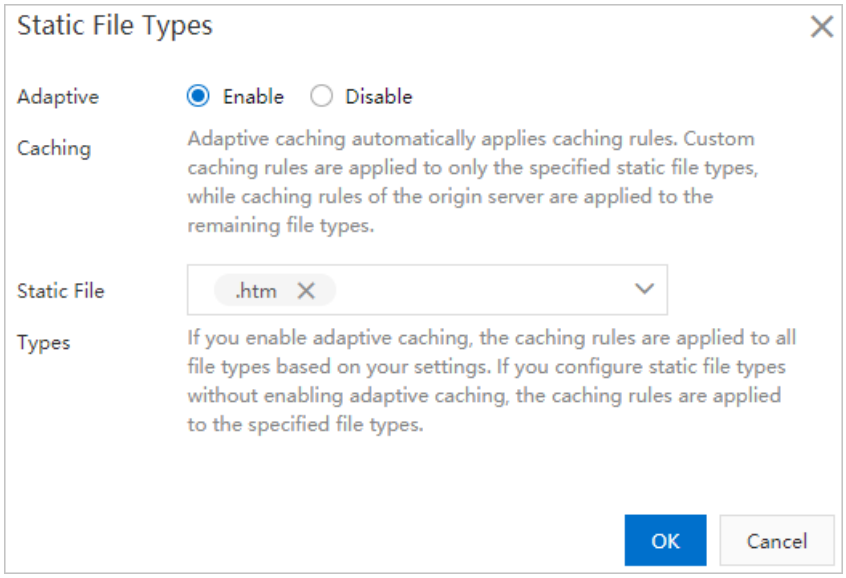
To accelerate the delivery of static and dynamic resources, you must turn on **Dynamic Acceleration**. You can configure acceleration rules for static file types based on your business requirements. Then, static and dynamic resources are distributed based on these custom acceleration rules. You can customize the file types, the Uniform Resource Identifiers (URIs), and the directories of static resources that can be cached on DCDN nodes.

- Disable dynamic acceleration

  If you do not want to accelerate the delivery of dynamic resources, you can turn off **Dynamic Acceleration**. After dynamic acceleration is disabled, the dynamic resources are distributed without acceleration. Static resources are distributed based on edge cache rules. Only the default static file acceleration rules are valid. All custom static file acceleration rules become invalid.

Procedure

1. Configure static file types.
   i.
   ii. In the left-side navigation pane, click **Domain Names**.
   iii. On the **Domain Names** page, find the domain name that you want to manage and click **Configure** in the Actions column.
   iv. The details page of the specified domain name appears. In the left-side navigation pane, click **Acceleration Rules**.
   v. Turn on **Dynamic Acceleration**.
   vi. On the **Static File Types** tab, click **Modify**.
   vii. In the **Static File Types** dialog box, choose to enable or disable **Adaptive Caching**, and then configure **Static File Types**.



Parameter	Description

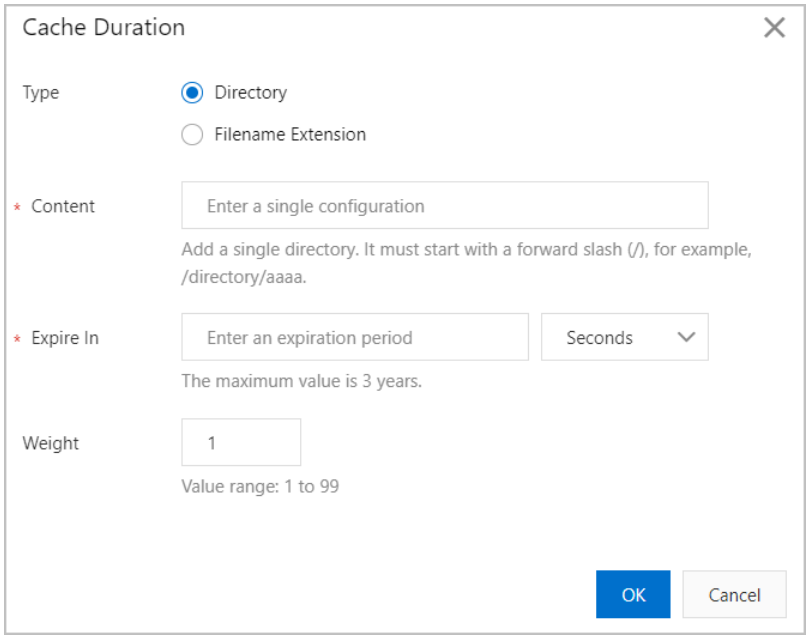
Parameter	Description
Adaptive Caching	You can configure the adaptive cache settings based on the static file type and the cache rule of the origin server. ■ Enable **Adaptive Caching** If you enable adaptive caching, rules in **Static File Types** take precedence over adaptive cache rules. All cache rules that are configured for **Cache Expiration** in the DCDN console take effect. Cache rules take effect in the following priorities: ■ Priority 1: If you have configured **Cache Expiration** in the DCDN console, the cache rules configured in **Cache Expiration** are applied. ■ Priority 2: If you have not configured **Cache Expiration** in the DCDN console, the cache rules that are configured on the origin server are applied. ■ Priority 3: If you have not configured cache rules in the DCDN console or on the origin server, DCDN dynamically initiates back-to-origin requests. ■ Disable **Adaptive Caching** ■ If you disable adaptive caching and have configured **Static File Types** and **Cache Expiration** in the DCDN console, the cache rules configured in **Cache Expiration** in the DCDN console are applied. **Note** Make sure that the filename extensions specified in **Cache Expiration** are included in **Static File Types**. Otherwise, the cache rules configured in **Cache Expiration** do not take effect. ■ If you disable adaptive caching and have configured **Static File Types** but have not configured **Cache Expiration**, rules that are configured on the origin server are applied.
Static File Types	The following static file types are supported: ■ Images: GIF, PNG, BMP, JPEG, and JPG. ■ Web pages: HTML, HTM, and SHTML. ■ Audio and video files: MP3, WMA, FLV, MP4, WMV, OGG, and AVI. ■ Text files: DOC, DOCX, XLS, XLSX, PPT, PPTX, TXT, and PDF. ■ Others: ZIP, EXE, TAT, ICO, CSS, JS, SWF, APK, M3U8, and TS.

viii. Click **OK**.

2. Create a cache expiration rule.

i. In the **Cache Expiration** section, click **Add**.

ii. In the **Cache Duration** dialog box, select **Directory** or **Filename Extension** for the Type parameter and specify the other required parameters for the cache rule.



iii. Click **OK**.

7.3. Specify static file URIs

Dynamic Route for CDN (DCDN) allows you to identify static files by Uniform Resource Identifier (URI). DCDN no longer uses dynamic acceleration to deliver the specified static files. Instead, DCDN uses static acceleration and allocates the optimal nodes for caching and delivery.

Context

The following acceleration rules are applicable to both static and dynamic resources:

- Enable dynamic acceleration

  To accelerate the delivery of static and dynamic resources, you must turn on **Dynamic Acceleration**. You can configure acceleration rules for static file types based on your business requirements. Then, static and dynamic resources are distributed based on these custom acceleration rules. You can customize the file types, the Uniform Resource Identifiers (URIs), and the directories of static resources that can be cached on DCDN nodes.

- Disable dynamic acceleration

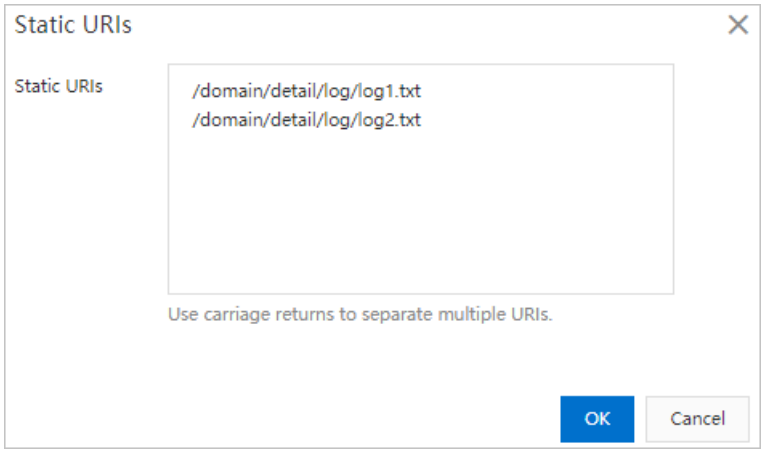
  If you do not want to accelerate the delivery of dynamic resources, you can turn off **Dynamic Acceleration**. After dynamic acceleration is disabled, the dynamic resources are distributed without acceleration. Static resources are distributed based on edge cache rules. Only the default static file acceleration rules are valid. All custom static file acceleration rules become invalid.

Procedure

1.
2.
3.

4.
5. Turn on **Dynamic Acceleration**.
6. On the **Static URIs** tab, click **Modify**.

7. In the **Static URIs** dialog box, specify **Static URIs**.



8.

7.4. Specify static file paths

Dynamic Route for CDN (DCDN) allows you to identify static files by file path. DCDN no longer uses dynamic acceleration to deliver the specified static files. Instead, DCDN uses static acceleration and allocates the optimal nodes for caching and delivery.

Context

The following acceleration rules are applicable to both static and dynamic resources:

- Enable dynamic acceleration

  To accelerate the delivery of static and dynamic resources, you must turn on **Dynamic Acceleration**. You can configure acceleration rules for static file types based on your business requirements. Then, static and dynamic resources are distributed based on these custom acceleration rules. You can customize the file types, the Uniform Resource Identifiers (URIs), and the directories of static resources that can be cached on DCDN nodes.

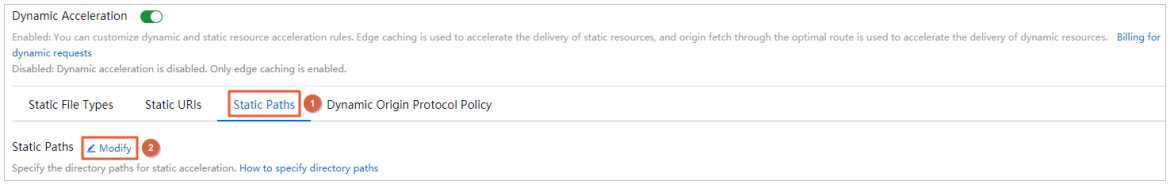
- Disable dynamic acceleration

  If you do not want to accelerate the delivery of dynamic resources, you can turn off **Dynamic Acceleration**. After dynamic acceleration is disabled, the dynamic resources are distributed without acceleration. Static resources are distributed based on edge cache rules. Only the default static file acceleration rules are valid. All custom static file acceleration rules become invalid.

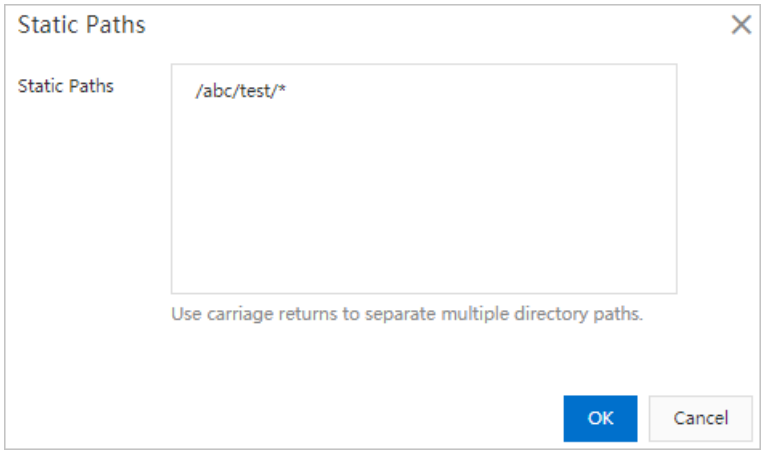
Procedure

1.

2.
3.
4.
5. Turn on **Dynamic Acceleration**.
6. On the **Static Paths** tab, click **Modify**.



7. In the **Static Paths** dialog box, specify **Static Paths**.



> **Note** You can use wildcards for a fuzzy search of static file paths. Asterisks (*) and question marks (?) can be used as wildcards. The asterisk (*) represents zero, one, or more characters, and the question mark (?) represents one character.

8.

7.5. Configure back-to-origin requests to retrieve dynamic content

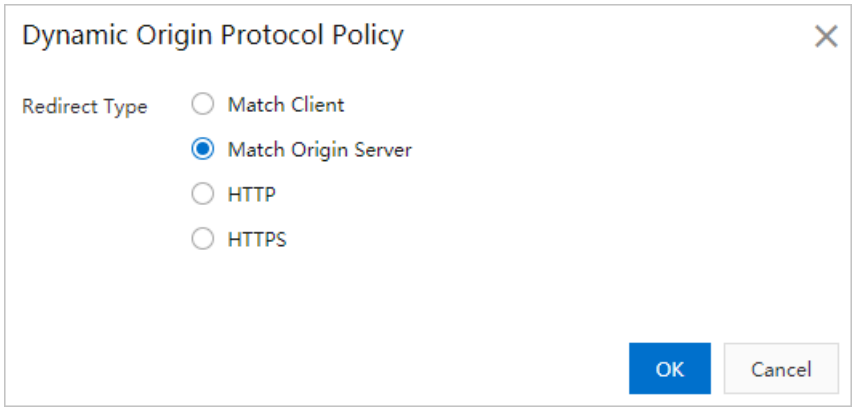
Dynamic Route for CDN (DCDN) accelerates the delivery of dynamic content by using an optimal route selected by an intelligent routing mechanism. This topic describes how to configure back-to-origin requests to retrieve dynamic content, including the protocol used to retrieve dynamic content and whether to enable load balancing. These configurations are independent of each other. You can perform configurations based on your business requirements.

Configure the origin protocol policy to retrieve dynamic content

The protocol used to retrieve dynamic content must be the same as that used by the client. If different protocols are used, you can configure the protocol used to retrieve dynamic content. If you do not configure the protocol, the protocol of the origin server port is used to retrieve dynamic content by default.

1.

2.
3.
4.
5. Turn on **Dynamic Acceleration**.
6. Click the **Dynamic Content Retrieval Settings** tab.
7. Click **Modify** next to **Dynamic Origin Protocol Policy**.
8. In the **Dynamic Origin Protocol Policy** dialog box, set **Redirect Type**.



Parameter	Description
Match Client	When the client uses HTTP or HTTPS to request resources, DCDN communicates with the origin server over the same protocol that is used by the client .
Match Origin Server	When the client uses HTTP or HTTPS to request resources, DCDN communicates with the origin server over the same protocol that is used by the origin server port.
HTTP	DCDN communicates with the origin server over HTTP.
HTTPS	DCDN communicates with the origin server over HTTPS.

9. Click **OK**.

Enable load balancing

Requests are redirected to origin servers based on the performance of origin servers. If you want requests to be redirected based on the weights of origin servers, you must enable load balancing. Then, requests are redirected to origin servers based on the specified weights to retrieve dynamic content.

- Performance-based redirection

  Requests are redirected to the origin server with the best performance.

- Weight-based redirection

  Requests are redirected to origin servers based on the specified weights. To modify the weight of an origin server, see Configure an origin server.

  1.
  2.

3.
4.
5. Turn on **Dynamic Acceleration**.
6. Click the **Dynamic Content Retrieval Settings** tab.
7. Turn on **Load Balancing** to enable **Load Balancing**.

8.Cache settings

8.1. Overview

When Alibaba Cloud DCDN accelerates the delivery of static resources to a user, it retrieves the resources from the origin server and caches them on the DCDN edge node that is nearest to the user. When the resources are requested again, the DCDN edge node directly returns the cached resources to the user. This accelerates content delivery. All the edge nodes of Alibaba Cloud DCDN are equipped with a caching system. When a user or an origin server interacts with an edge node, the caching system responds to user requests or processes responses from the origin server. For example, the caching system specifies a time-to-live (TTL) for cached resources or rewrites back-to-origin requests.

DCDN supports the following caching features.

Feature	Description
Add a cache rule for resources	Allows you to configure cache expiration rules for static resources in a specified directory or with specified file extensions. In each cache expiration rule, you can set a TTL for the cached static resources and a priority for the rule. DCDN edge nodes cache and expire static resources based on the cache expiration rules.
Create a cache expiration rule for HTTP status codes	Allows you to set a TTL for HTTP status codes that are returned to requests for resources in a specified directory or with specified file extensions.
Customize an HTTP header	Allows you to customize HTTP response headers for expired resources.
Customize an error page	Allows you to customize an error page for a specific HTTP status code.
Create a URI rewrite rule	Allows you to redirect request URIs to the specified URIs by using 302 redirects.

8.2. Add a cache rule for resources

Cache duration refers to the amount of time that a resource is cached on Alibaba Cloud Dynamic Route for CDN (DCDN) nodes. When the cache duration of a cached resource ends, the resource on the DCDN nodes expires. Requests that attempt to access expired resources are redirected to the origin server. The retrieved resources are returned to the clients and cached on the DCDN nodes. You can add a cache rule for static resources based on file directories or filename extensions.

This topic consists of the following sections:

- Usage notes
- Procedure
- Priorities of DCDN cache rules
- HTTP caching mechanisms
- Configuration examples
- Related API operations

Usage notes

- When you update a file on the origin server, we recommend that you add version numbers to the name of the file. This way, you can differentiate between file versions.

  To differentiate file versions after updates, we recommend that you add version numbers to file names. This way, each file version has a unique name. For example, you can name a file *img-v1.0.jpg* before it is updated and *img-v2.1.jpg* after it is updated.

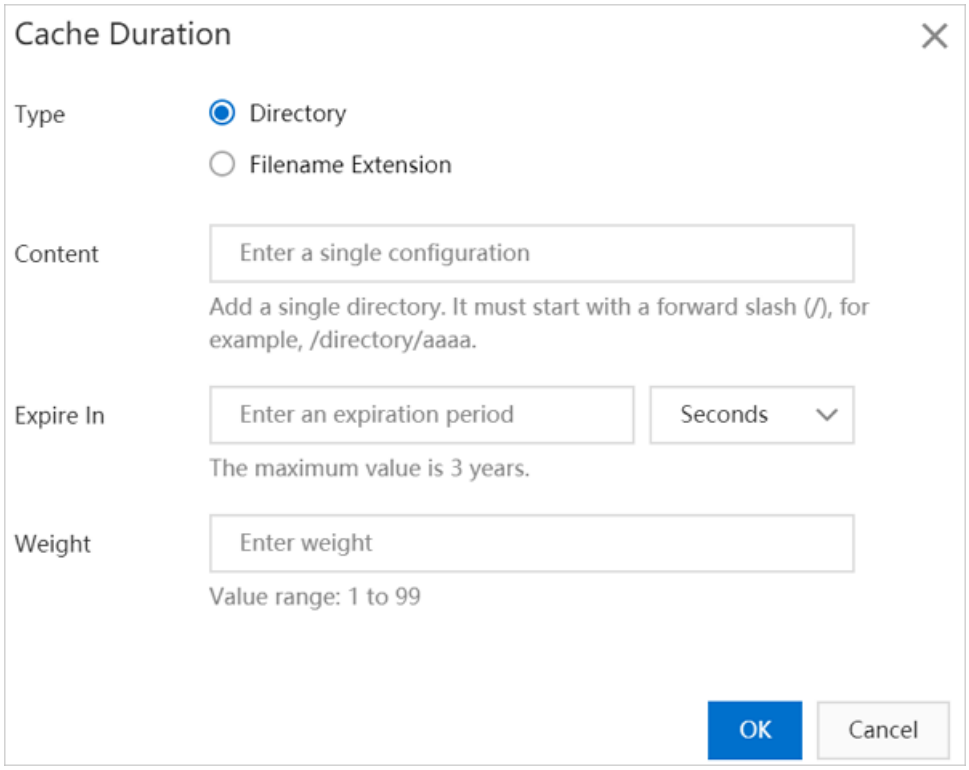
- The resource cache duration affects the back-to-origin routing frequency. Set a proper cache duration based on your business requirements.

  A short cache duration may cause frequent back-to-origin routing and increase loads on the origin server. A long cache duration may cause resources on DCDN nodes to be outdated.

- Cached resources that are infrequently requested may be removed from the DCDN nodes before they expire.
- If a DCDN node retrieves a static file from an origin server, the node processes the file based on the priorities of the cache rules. For more information, see Priorities of DCDN cache rules.

Procedure

1.
2.
3.
4.
5. On the **Cache Duration** tab, click **Add**.
6. In the **Cache Duration** dialog box, configure the parameters that are described in the following table.



Parameter	Description
Type	You can select **Directory** or **Filename Extension**. ○ **Directory**: Add a cache rule for resources under the specified directory. ○ **Filename Extension**: Add a cache rule for resources with the specified filename extension.
Content	Specify the directories or filename extensions for which you want to add the cache rule. ○ If you select **Directory**, take note of the following rules: ■ You can enter only one directory at a time. You can use a forward slash (/) to specify all directories. ■ You can enter a full path. The path must start with a forward slash (/), for example, */directory/aaa*. ○ If you select **Filename Extension**, take note of the following rules: ■ You can enter one or more filename extensions and separate them with commas (,), for example, `jpg,txt`. Filename extensions are case-sensitive. The following static file types are supported: ■ Images: GIF, PNG, BMP, JPEG, and JPG. ■ Web pages: HTML, HTM, and SHTML. ■ Audio and video files: MP3, WMA, FLV, MP4, WMV, OGG, and AVI. ■ Text files: DOC, DOCX, XLS, XLSX, PPT, PPTX, TXT, and PDF. ■ Others: ZIP, EXE, TAT, ICO, CSS, JS, SWF, APK, M3U8, and TS. ■ You cannot use a wildcard character (*) to specify all file types.
Expire In	The cache duration for cached resources. The maximum cache duration is three years. Take note of the following rules: ○ Specify a cache duration of one month or longer for static files that are infrequently updated, such as images and application packages. ○ Specify a cache duration based on your business requirements for static files that are frequently updated, such as JS and CSS files. ○ We recommend that you specify a cache duration of 0 seconds to disable caching for dynamic files, such as PHP, JSP, and ASP files.

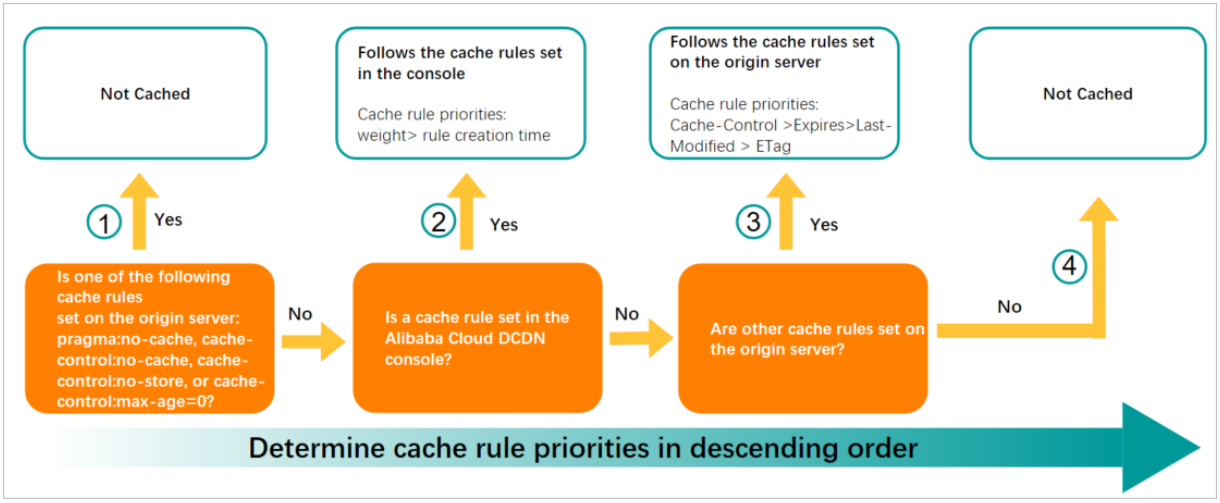
Parameter	Description
Weight	The weight for a cache rule, which indicates the priority of the cache rule. Valid values are 1 to 99. A greater value indicates a higher priority. **Note** • If you create multiple cache rules, we recommend that you set a unique weight for each cache rule to define their priorities. • Cache rules that have the same weight are prioritized in order of creation time, regardless of the rule type. The rule with the earliest creation time applies. • If you have configured multiple cache rules for the same cached resource, only the first matched rule is applied.

7. Click **OK**.

   After you add a cache rule, you can **Modify** or **Delete** the cache rule on the **Cache Duration** tab.

Priorities of DCDN cache rules

After a DCDN node retrieves a static file from an origin server, the node processes the static file based on the cache rules in the following order. A smaller value indicates a higher priority.



1. If the response carries the `pragma:no-cache`, `cache-control:no-cache`, `cache-control:no-store`, or `cache-control:max-age=0` directive, the static file is not cached.
2. DCDN follows the cache duration for cached resources, or the validity period for HTTP status codes set in the DCDN console.

> **Note**
>
> If a request matches multiple cache rules, only one rule is applied in the following order of priority: weight > rule creation time.
>
> - If you create multiple cache rules, we recommend that you set a unique weight for each cache rule to define their priorities. A higher weight indicates a higher priority.
> - Cache rules that have the same weight are prioritized in order of creation time, regardless of the rule type. The rule with the earliest creation time applies.

3. DCDN follows other cache rules set on the origin server. Headers in responses from the origin server are in the following descending order of priority: Cache-Control > Expires > Last-Modified > ETag.
   i. The response carries the `Cache-Control` header and the directive is `max-age`, which is set to a value greater than 0, for example, `cache-control:max-age=3600`.
   ii. The response carries the `Expires` header, for example, expires:Tue, 25 Nov 2031 17:25:43 GMT.
   iii. If the response carries the `ETag` or `Last-Modified` header, the cache duration is calculated based on the following rules:
      a. If the response carries the `Last-Modified` header, the cache duration is calculated based on the following formula: Cache duration = (Current time - Last-Modified) × 0.1. If the result is between 10 seconds to 3,600 seconds, the result applies. If the result is less than 10 seconds, the cache duration is 10 seconds. If the result is greater than 3,600 seconds, the cache duration is 3,600 seconds.
      b. If the response carries only the `ETag` header, the cache duration is 10 seconds.
4. If the response does not carry the `ETag`, `Last-Modified`, `Cache-Control`, or `Expires` header, the static file is not cached on the node.

HTTP caching mechanisms

HTTP provides three types of headers that can be used to control caching behaviors.

1. **Cache duration**

   When a client requests resources from a server, the client and server define the cache duration of the returned resources that are cached on DCDN nodes. The resources expire when the cache duration ends.

   HTTP provides the following types of headers that can be used to define the cache duration.

Header	Protocol version	Description	Example	Type
Pragma	HTTP/1.0	The Pragma header specifies whether a resource is cached. If Pragma is set to no-cache, the resource is not cached. Pragma is compatible with servers that use only HTTP/1.0.	Pragma:no-cache	Request and response

Header	Protocol version	Description	Example	Type
Expires	HTTP/1.0	The Expires header specifies a date and time. The cached resource expires at the specified time. If Expires is set to an invalid date, such as 0, the resource has already expired.	Expires: Wed, 21 Oct 2022 07:28:00 GMT	Response
Cache-Control	HTTP/1.1	The Cache-Control header can be set to different directives to control the caching behaviors. Most mainstream clients, such as browsers, use Cache-Control to control the caching behaviors.	The following directives specify that files are not cached: ◦ Cache-Control:no-cache ◦ Cache-Control:no-store ◦ Cache-Control:max-age=0 The following directive specifies that files are cached for 1 hour: Cache-Control:max-age=3600.	Request and response

2. **Resource tags**

   The first time a client requests a resource from a server, the server adds a tag to the response headers. The next time the client requests the resource from the server, the tag is used to identify the requested resource. The header of the subsequent requests for the same resource carries this tag. If the server checks the tag and confirms that the requested resource is not updated, the HTTP 304 status code is returned to the clients. The clients retrieve the resource from the local cache. If the server detects that the tag is different from that of the resource on the server, the server informs the clients that the resource has been updated or has expired. In this case, the clients must retrieve the latest version of the resource from the server.

   HTTP provides the following types of headers that can be used to control cache versions.

Header	Protocol version	Description	Example	Type
Last-Modified	HTTP/1.0	Last-Modified indicates the time when a resource was last updated.	Last-Modified: Wed, 21 Oct 2015 07:28:00 GMT	Response

Header	Protocol version	Description	Example	Type
ETag	HTTP/1.1	The ETag header is the unique identifier of each version of a resource. ETag indicates whether a resource has been updated. If the resource has been updated, the server does not need to return a complete response.	ETag: "33a64df551425fcc55e4d42a148795d9f25f89d4"	Response

3. **Content negotiation**

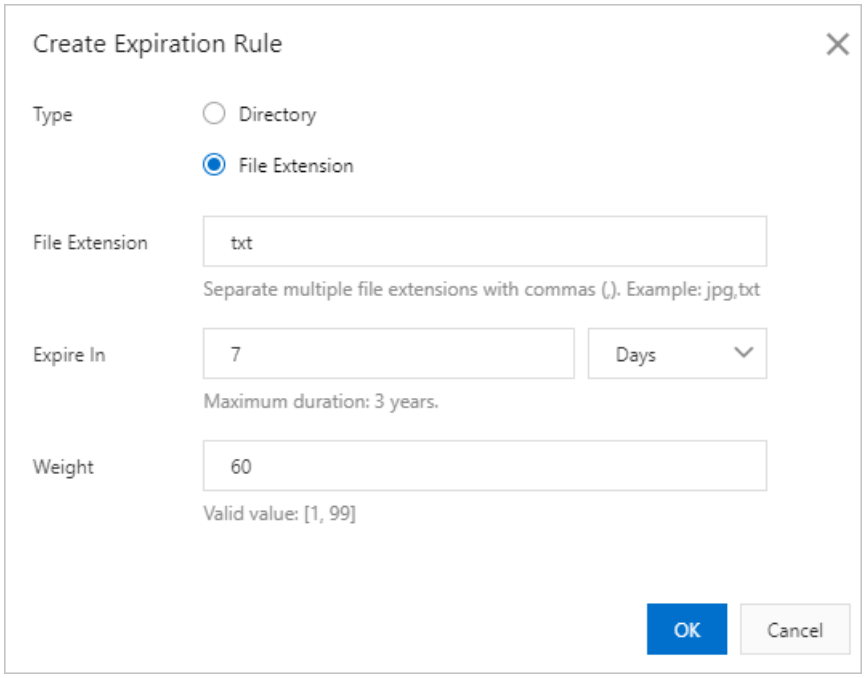
   Caching software uses keywords to index objects on disks. In HTTP/1.0, URLs are used as keywords. However, different resources may point to the same URL. To differentiate them, clients must provide more information, such as the Accept-Language and Accept-Charset headers. HTTP/1.1 introduced the Vary response header to implement content negotiation. The Vary header lists the request headers that must be included to implement content negotiation.

   In content negotiation, Vary is used to differentiate versions of the requested resource so that the clients can retrieve the desired version of the requested resource.

Header	Protocol version	Description	Example	Type
Vary	HTTP/1.1	Examples ○ A server uses `Vary: Accept-Encoding` to inform the recipient, such as a DCDN node, that the requested resource has two versions. One is compressed, and the other is not. When the client sends requests to DCDN for the same resource, the browser with an earlier version requires the resource to be uncompressed to prevent incompatibility. The browser with the latest version requires the resource to be compressed to reduce data transfer. ○ The server uses `Vary: User-Agent` to identify the browsers that initiate the requests and inform the recipient, such as a DCDN node, of the browser types. The DCDN node caches the resource of the required version based on the browser types.	Vary: Accept-Encoding Vary: Accept-Encoding,User-Agent	Response

Configuration examples

Example 1: If you want DCDN nodes to cache .txt files for 7 days, add a cache rule for .txt files in the DCDN console and set the cache duration to 7 days.



Example 2: The following cache rules are set for the accelerated domain name `demo.aliyun.com`. When DCDN nodes retrieve the resource `http://demo.aliyun.com/image/example.png`, each of the rules is a match. In addition, the rules have the same weight. In this case, the rules are prioritized in order of creation time. The rule with the earliest creation time has the higher priority. Therefore, the rule that is set for the /image directory is applied because the rule is created the earliest.

Object	Type	Expire In	Weight	Status	Actions
/image	Directory	1 Days	10	Successful	Modify | Delete
jpg,png	File Extension	1 Months	10	Successful	Modify | Delete

Related API operations

BatchSetDcdnDomainConfigs

8.3. Create a cache expiration rule for HTTP status codes

This topic describes how to set a time-to-live (TTL) value for HTTP status codes that are returned to requests for static resources in a specified directory or with specified file extensions. This allows DCDN edge nodes to directly return HTTP status codes to requests and reduces loads on origin servers.

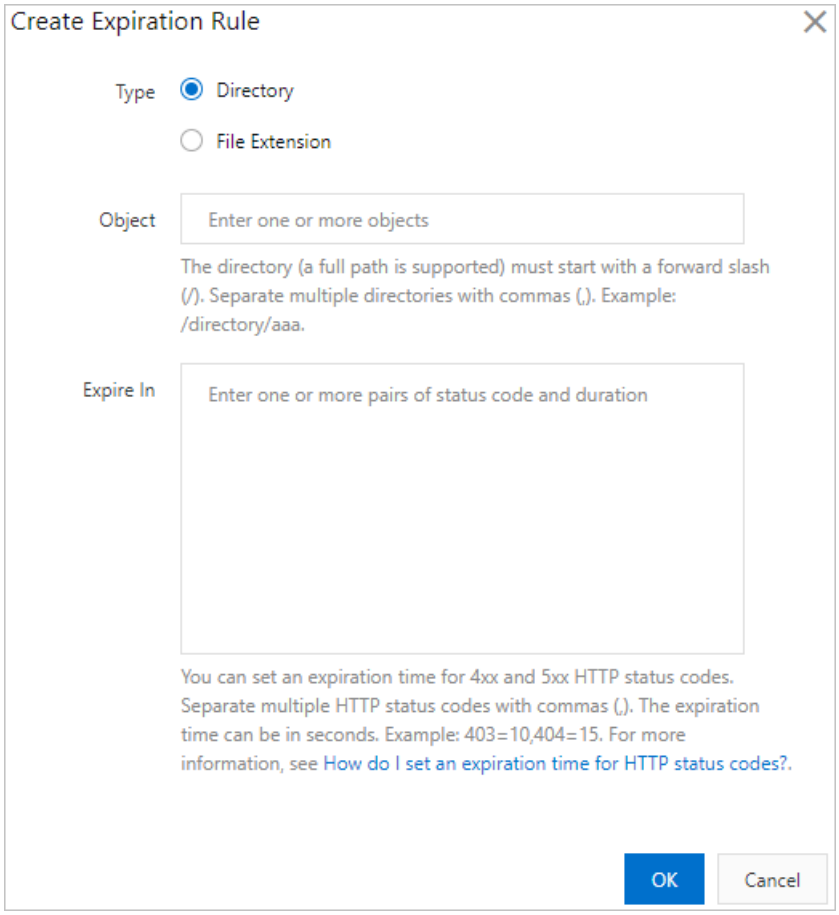
Scenarios

HTTP 2xx status codes indicate that DCDN edge nodes have retrieved the requested resources from the origin server. In this case, the origin server returns an HTTP 2xx status code to the DCDN edge nodes. Then, the DCDN edge nodes process the requests based on cache expiration rules. If the origin server fails to return HTTP status codes, such as an HTTP 2xx status code, to the DCDN edge nodes, and you do not want the origin server to respond to every request, you can set a TTL value for HTTP status codes. Then, DCDN edge nodes directly return HTTP status codes to requests to reduce loads on the origin server.

> **Note**
>
> - HTTP 303, 304, 401, 407, 600, and 601 status codes are not supported.
> - If the Cache-Control, Pragma or Expires header is configured on the origin server, Alibaba Cloud DCDN caches the following HTTP status codes based on the Cache-Control, Pragma or Expires directive: 204, 305, 400, 403, 404, 405, 414, 500, 501, 502, 503, and 504. If you do not set a TTL value for these HTTP status codes, or the HTTP response header of the origin server does not contain the field Cache-Control, Pragma or Expires, the TTL value is set to 1 second by default.

Procedure

1.
2.
3.
4.
5. Click the **Status Codes and Expiration Time** tab.
6. Click **Add** to create an expiration rule for HTTP status codes.



Parameter	Description
Type	You can select **Directory** or **File Extension**. Select a type based on your business requirements. **Note** If you set TTL values for the HTTP status codes of both a directory and files with specified extensions, whichever rule that is set first takes effect. All other rules are ignored.
Object	◦ If you select **Directory**, take note of the following rules: ■ You can add only one directory in each rule. ■ You can enter a full path. It must start with a forward slash (/), for example, */directory/aaa*. ◦ If you select **File Extension**, take note of the following rules: ■ You can enter one or more file extensions. Separate file extensions with commas (,), for example, `JPG,TXT`. ■ You cannot use an asterisk (*) to specify all file types.

Parameter	Description
Expire In	○ You can set a TTL value for 4xx and 5xx HTTP status codes. The TTL value is in seconds. Separate HTTP status codes with commas (,), for example, 4xx=10,5xx=15. ○ You cannot set a TTL value for 2xx or 3xx status codes. You can set a TTL value only for specific HTTP status codes, including 201 and 302 status codes. The TTL value is measured in seconds. For example, you can set 201=10,302=15.

7. Click **OK**.

   After an expiration rule for HTTP status codes is created, it is displayed on the **Status Code Expiration Time** tab. You can **Modify** or **Delete** the rule.

Related API operations

BatchAddDcdnDomain

8.4. Customize an HTTP header

HTTP headers define the resources that are being requested, the behaviors of clients or servers, and the operation parameters of HTTP requests. This topic describes how to customize an HTTP response header.

Context

HTTP headers are the components of header sections in request and response messages that are transferred over HTTP.

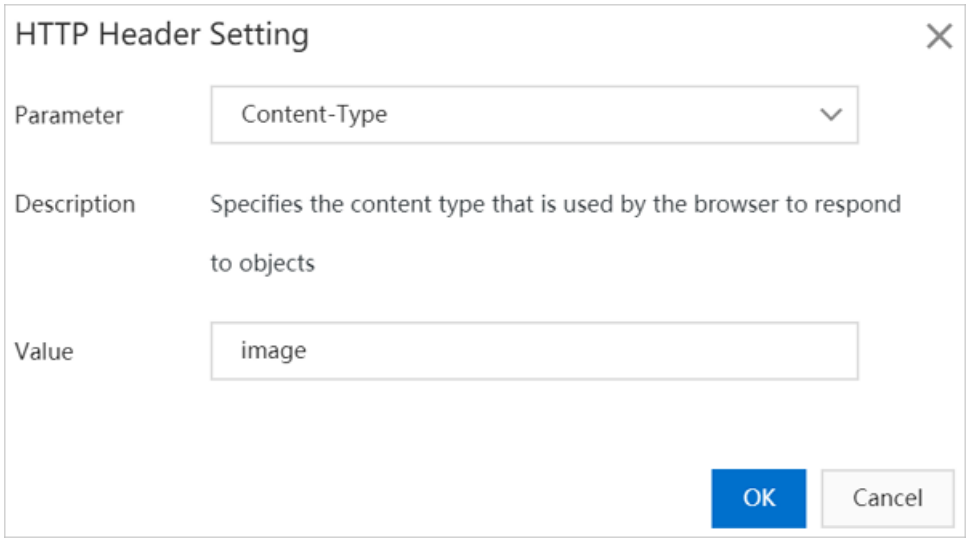
HTTP headers include general headers, request headers, and response headers.

When you create an HTTP response header, pay attention to the following notes:

- The configurations of the HTTP response header for an accelerated domain name affect how a client program responds to all the requests that are destined for the domain. The client program can be a browser. The header configurations do not affect the cache server.
- You can set the value of `Access-Control-Allow-Origin` to an asterisk (`*`) to specify all the domain names. You can also set the value to a specific domain name, such as `www.aliyun.com` .
- DCDN does not support configuring response headers for wildcard domains.

Procedure

1.
2.
3.
4.
5. On the **HTTP Header** tab, click **Add**.
6. In the **Custom HTTP Response Header Settings** dialog box, select an HTTP header from the Parameter drop-down list and specify the Value parameter.



The following table describes the 10 HTTP response headers that are provided by Dynamic Route for CDN (DCDN). If you need to specify other HTTP response headers, submit a ticket.

Parameter	Description	Example
Content-Type	Specifies the type of the content that is returned to the client program.	image
Cache-Control	Specifies the cache policy that the client program follows for requests and responses.	no-cache
Content-Disposition	Specifies the default file name that is provided by the client program when the requested content is saved as a file.	123.txt
Content-Language	Specifies the language of the intended audience for the content that is returned to the client program.	zh-CN
Expires	Specifies the expiration time of the content that is returned to the client program.	Wed, 21 Oct 2015 07:28:00 GMT
Access-Control-Allow-Origin	Specifies the origins from which cross-origin requests are allowed.	* **Note** You can enter `*` in the Value field to specify all the domain names. You can also enter a full domain name, such as `www.aliyun.com`.

Parameter	Description	Example
Access-Control-Allow-Headers	Specifies the fields that are allowed in cross-origin requests.	X-Custom-Header
Access-Control-Allow-Methods	Specifies the request methods that are allowed for cross-origin requests.	POST, GET Note To add the POST method and the GET method, separate them with a comma (,).
Access-Control-Max-Age	Specifies the time to live (TTL) during which the response can be cached for a preflight request that is initiated by the client program for a particular resource.	600
Access-Control-Expose-Headers	Specifies the headers that can be exposed as part of the response.	Content-Length

7. Click **OK**.

   On the **HTTP Header** page, you can click **Modify** or **Delete** to manage the HTTP header.

8.5. Customize an error page

When a client requests a web service through a browser, the website hosting server returns the default 404 Not Found page if the requested URL does not exist. The default error page of a web server may not meet your requirement. To improve user experience, you can associate full URLs with error codes that are included in HTTP or HTTPS responses. When an error occurs, the server returns the associated custom page. This topic describes how to customize an error page.

Context

Alibaba Cloud provides the default page and the custom page for HTTP error codes. The 404 error code is taken as an example to describe the differences between the default page and the custom page.

- Default page: When the HTTP response includes the 404 status code, the server returns the default 404 Not Found page.
- Custom page: When the HTTP response includes the 404 status code, the server returns the custom page. You must specify a full URL for the custom page.

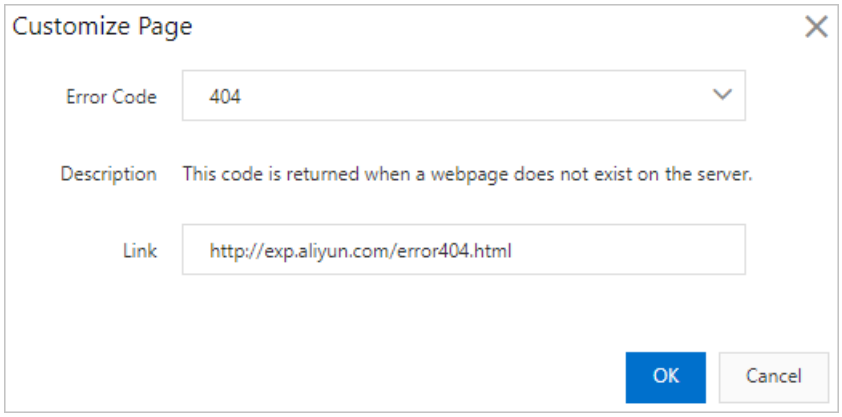
  Note Custom pages are considered as personal resources and fees are charged based on the specified billing rules.

Procedure

1.

2.
3.
4.
5. Click the **Custom Pages** tab.
6. On the **Custom Pages** tab, click **Add**.
7. In the **Custom Pages** dialog box, specify the **Error Code** and **Link** parameters.

   For example, you want to store the `error404.html` page and other static files on the origin server. You also want to return this error page to requests that are destined for the accelerated domain name `exp.aliyun.com`. Then, you only need to select **404** from the Error Code drop-down list and enter the full URL `http://exp.aliyun.com/error404.html` in the Link field.



8. Click **OK**.

   On the **Custom Pages** page, you can click **Modify** or **Delete** to manage the custom page.

8.6. Create a URI rewrite rule

You can create a URI rewrite rule to redirect requests from HTTP URIs to HTTPS URIs, or rewrite URIs in requests to the URIs of the requested resources.

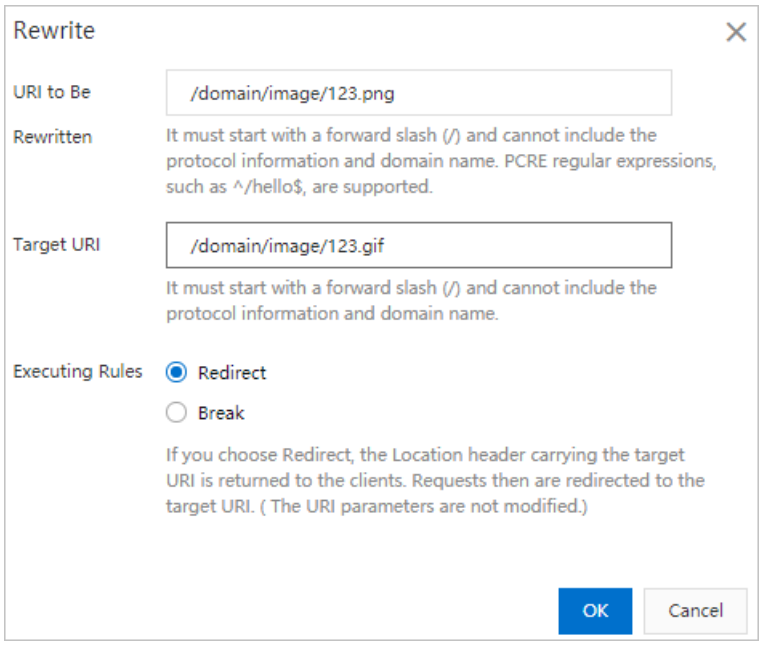
Scenarios

If you want to rewrite URIs in requests to the URIs of the requested resources, you can create a URI rewrite rule. If a request matches the rewrite rule, Alibaba Cloud Dynamic Route for CDN (DCDN) performs a 302 redirect to redirect the request to the final URI. For example, users visit `www.example.com/hello` over HTTP. After you create a rewrite rule, requests that are sent to `www.example.com/hello` are redirected to `www.example.com/index.html`.

Procedure

1.
2.
3.
4.
5. Click the **URI Rewrite** tab.
6. Click **Add** and configure **URI to Be Rewritten**, **Target URI**, and **Executing Rules** based on your business requirements.

> **Note** You can create up to 50 rewrite rules for each domain name.



Parameter	Description
URI to Be Rewritten	Enter a URI that starts with a forward slash (/). The URI cannot contain http:// or domain names. You can use Perl Compatible Regular Expressions (PCRE) to specify the URI, for example, ^/hello$.
Target URI	Enter a URI that starts with a forward slash (/). The URI cannot contain http:// or domain names. Example: /index.html.
Executing Rules	**Redirect** and **Break** are supported. ○ **Redirect**: If the URI in a request matches the current rule, Alibaba Cloud DCDN performs a 302 redirect to redirect the request to the final URI. ○ **Break**: If the URI in a request matches the current rule, Alibaba Cloud DCDN returns the content of the final URI and skips other rewrite rules.

7. Click **OK**.

   After a rewrite rule is created, it is displayed on the **URI Rewrite** tab. You can **Modify** or **Delete** the rule.

Configuration examples

Example	URI to Be Rewritten	Target URI	Executing Rules	Description
Example 1	/hello	/index.html	Redirect	A client requests `example.aliyundoc.com/hello`. DCDN nodes perform a 302 redirect to redirect the request to `example.aliyundoc.com/index.html`.

Example	URI to Be Rewritten	Target URI	Executing Rules	Description
Example 2	^/$	/index.html	Redirect	A client requests `example.aliyundoc.com`. DCDN nodes perform a 302 redirect to redirect the request to `example.aliyundoc.com/index.html`.
Example 3	/hello	/hello/index.html	Redirect	A client requests `example.aliyundoc.com/hello`. DCDN nodes perform a 302 redirect to redirect the request to `example.aliyundoc.com/hello/index.html`.
Example 4	^/hello$	/index.html	Break	A client requests `example.aliyundoc.com/hello`. DCDN nodes return the content of `example.aliyundoc.com/index.html` and skip the subsequent rules.

API operations

BatchSetDcdnDomainConfigs

9.HTTPS settings

9.1. What is HTTPS secure acceleration?

This topic describes the benefits and usage notes of HTTPS secure acceleration and how it works. HTTPS secure acceleration is used to encrypt connections between clients and Dynamic Route for CDN (DCDN) nodes. HTTPS ensures data security during transmission.

This topic consists of the following sections:

- What is HTTPS?
- How it works
- Billing
- Benefits
- Scenarios
- Usage notes
- Enable HTTPS secure acceleration

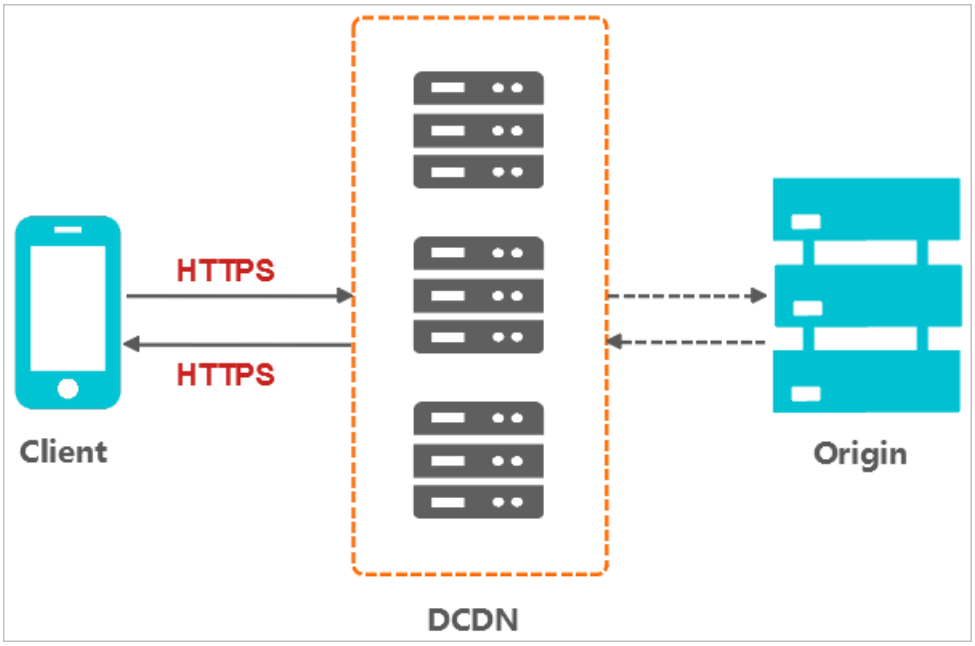
What is HTTPS?

HTTP transmits data in plaintext and does not encrypt data. HTTPS is an extension of HTTP and is designed to ensure data security. In HTTPS, the communication protocol is encrypted by using Transport Layer Security (TLS), formerly known as Secure Sockets Layer (SSL). HTTPS is used to encrypt connections. HTTPS is widely used to protect sensitive user data for services such as payment transactions.

How it works

After you enable HTTPS in the DCDN console, transmissions between clients and DCDN nodes are encrypted over HTTPS. If you want to enable end-to-end HTTPS encryption, you must configure DCDN nodes to redirect requests to origin servers over HTTPS. Make sure that the origin servers support HTTPS.

The following figure shows how HTTPS works.



1. In the Alibaba Cloud DCDN console, configure the public and private keys of the SSL certificate on DCDN nodes.

   > **Note** You can acquire the public and private keys by applying for or uploading an SSL certificate.

2. The DCDN node sends the public and private keys to the client.
3. The client parses the public key to verify the validity.
   - If the public key is valid, the client generates a random number. The client uses the public key to encrypt the random number and transmits the number to the DCDN node.
   - If the public key is invalid, SSL handshakes fail. You must configure a valid SSL certificate.

   > **Note** A certificate is considered valid only if the following requirements are met:
   > - The certificate is not expired.
   > - The certificate is issued by a trusted certificate authority (CA).
   > - The public key of the certificate can decrypt the certificate signature signed by the CA.
   > - The domain name on the certificate matches the accelerated domain name.

4. The DCDN node uses the private key to decrypt the encrypted random number.
5. The DCDN node uses the random number to encrypt data transmission.
6. The client uses the random number to decrypt the received data.

Billing

HTTPS secure acceleration is a value-added service. After you enable HTTPS, you are charged based on the number of HTTPS requests. For more information, see Billing of requests and WebSocket.

> **Note** HTTPS requests are separately billed, and the fees cannot be offset by data transfer plans of DCDN. Make sure that you have a sufficient balance in your Alibaba Cloud account. Otherwise, overdue payments may occur and cause service suspension.

Benefits

HTTPS secure acceleration provides the following benefits:

- HTTPS secure acceleration protects communications from eavesdropping, tampering, impersonation attacks, and man-in-the-middle (MITM) attacks.
- HTTPS encrypts sensitive information such as session IDs and cookies before transmission. This minimizes the risk of sensitive information leaks.
- HTTPS checks data integrity during transmission to protect the data from MITM attacks, such as DNS hijacking and tampering.
- HTTPS is the new standard. An increasing number of mainstream browsers such as Google Chrome 70 and later and Mozilla Firefox have labeled HTTP web URLs as not secure since 2018. If you choose to use HTTP, your website may be exposed to security risks. Users who visit your website by using these browsers are prompted that this website is not secure. This compromises user experience and may reduce visits to the website.
- Mainstream search engines have a higher weight for HTTPS-capable websites. After you enable HTTPS for your website, the website can achieve a higher ranking in search engine optimization (SOE). HTTP/2 is supported by a growing number of browsers because HTTP/2 can provide a better user experience. A website must support HTTPS before it can support HTTP/2. HTTPS is a more reliable choice in terms of security, market presence, and user experience. Therefore, we recommend that you upgrade your communication protocol to HTTPS.

Scenarios

The following table describes the use scenarios of HTTPS secure acceleration.

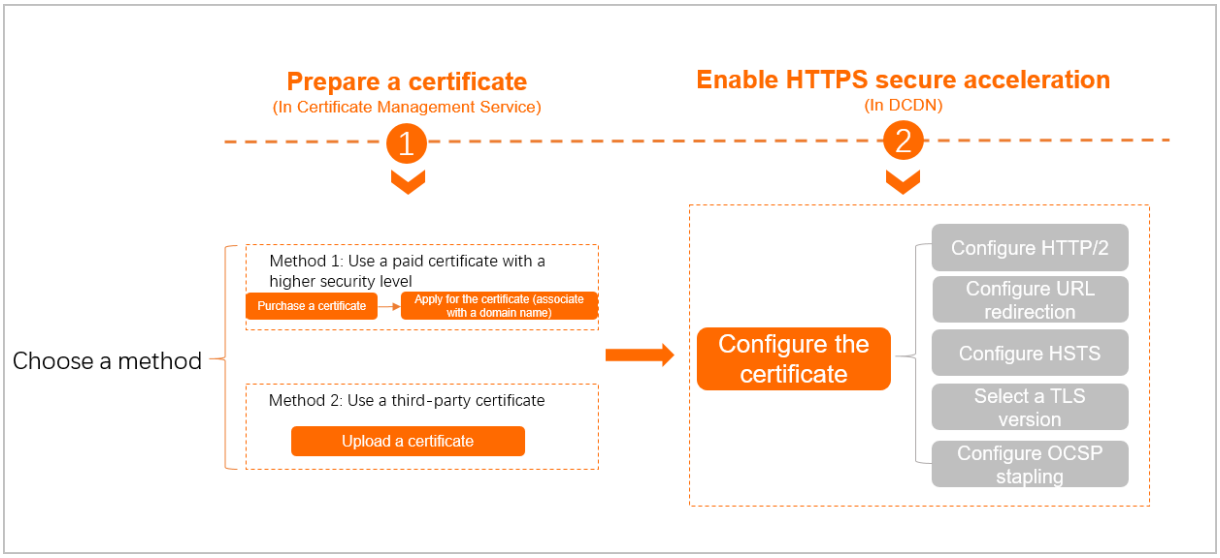
Scenario	Description
Enterprise applications	HTTPS protects confidential information on enterprise websites from being hijacked or intercepted. Leaks of the confidential information, such as customer relationship management (CRM) data and enterprise resource planning (ERP) data, may cause fatal damages to enterprises.
Public service websites	HTTPS protects sensitive information on public service websites against attacks such as phishing and hijacking. Leaks of such information may compromise public trust.
Payment systems	HTTPS protects sensitive data such as customer names and phone numbers used in payment transactions against hijacking and spoofing. If sensitive data is leaked, attackers can use such data to trick customers into making duplicate payments. This causes losses to both the customer and the enterprise.
API operations	API operations can use HTTPS to encrypt important information, such as sensitive data and important instructions. This protects the information against hijacking.
Enterprise websites	HTTPS improves user trust and experience. Web browsers display a lock icon in the address bar for websites with domain validated (DV) or organization validated (OV) certificates. The enterprise name is displayed together with the lock icon for websites that include extended validated (EV) certificates.

Usage notes

The following table describes the usage notes of HTTPS.

Category	Note
Scenario	• All domain names can enable HTTPS regardless of the content type. • You can enable HTTPS for a wildcard domain name. • You can renew an SSL certificate. Proceed with caution. After an SSL certificate is renewed, it takes effect within one minute. Usage notes on enabling and disabling HTTPS: • Enable HTTPS: After you enable HTTPS, you can change SSL certificates. You can also configure URL redirection to redirect user requests from HTTP to HTTPS. For more information, see Configure force redirect. • Disable HTTPS: After you disable HTTPS, the system no longer supports HTTPS requests and retains the SSL certificate or private key information. If you want to enable HTTPS again, you must select an SSL certificate from Certificate Management Service. For more information, see Configure an SSL certificate.
Billing	HTTPS secure acceleration is a value-added service. After you enable HTTPS, you are charged based on the number of HTTPS requests. For more information, see Billing of requests and WebSocket. **Note** HTTPS requests are separately billed, and the fees cannot be offset by data transfer plans of DCDN. Before you enable HTTPS secure acceleration, make sure that you have a sufficient balance in your Alibaba Cloud account. If the balance is insufficient, DCDN may be suspended.
Certificate management	• You must upload SSL certificates and private keys in `Privacy-Enhanced Mail (PEM)` format for domain names for which you want to enable HTTPS secure acceleration. **Note** The Tengine web server used by DCDN is designed based on the NGINX web server architecture. Therefore, the web server supports only certificate files in NGINX-compatible `PEM` format. For more information, see Certificate formats. • The uploaded SSL certificate must match the private key. Otherwise, requests sent from clients fail the authentication. • The system does not support the private keys for which passwords are configured. • Only SSL and TLS handshakes that include Server Name Indication (SNI) values are supported. • You can view SSL certificates. You cannot view private keys because they are sensitive information. Keep certificate-related information confidential.

Enable HTTPS secure acceleration



1. Prepare an SSL certificate in the Certificate Management Service console.

   The following types of SSL certificates are supported. Select a type based on your business requirements and configure an SSL certificate in the Certificate Management Service console.

   - Premium certificates (with higher security): Purchase a certificate > Submit a certificate application.
   - Third-party certificates: You must upload the certificate to Certificate Management Service. For more information, see Upload a certificate.

2. Enable HTTPS in the DCDN console.

   i. After you prepare an SSL certificate, you must configure the certificate before HTTPS can be enabled. For more information, see Configure an SSL certificate.

   ii. (Optional) You can configure advanced features such as Force Redirect based on your business requirements.

Feature	Description
Enable HTTP/2	HTTP/2 is a binary protocol developed based on HTTP/1.1. HTTP/2 significantly improves web performance and reduces latency by enabling multiplexing and header compression. HTTP/2 is supported by mainstream browsers, including Google Chrome, Microsoft Edge, Safari, and Mozilla Firefox.
Configure force redirect	Redirects requests to HTTP or HTTPS.
Configure HSTS	Forces clients such as browsers to communicate with servers over HTTPS. This reduces the risk of cookie hijacking.
Configure TLS version control	Ensures communication security and data integrity.
Configure OCSP stapling	Caches the revocation status of SSL certificates and returns the information to clients. Clients do not need to query the revocation status of SSL certificates from certificate authorities (CAs). This reduces the verification time.

9.2. Certificate formats

To access resources over HTTPS secure acceleration, you must configure an SSL certificate. This topic describes the certificate formats that are supported by Alibaba Cloud Dynamic Route for CDN (DCDN) and how to convert certificate formats.

Root CA certificates

Root CA certificates are issued by root certificate authorities (CAs) including Apache, IIS, NGINX, and Tomcat. Each root CA certificate is unique. Alibaba Cloud DCDN uses root CA certificates that are issued by NGINX. The certificate information is contained in a `.crt` file and the private key is contained in a `.key` file.

Take note of the following rules when you upload a certificate:

- The certificate must start with `-----BEGIN CERTIFICATE-----` and end with `-----END CERTIFICATE-----`.
- All lines except the last line must be 64 characters in length. The last line can be up to 64 characters in length.

The following figure shows a sample certificate in `PEM` format that is used in Linux.

Certificates issued by an intermediate CA

A certificate file that is issued by an intermediate CA contains multiple certificates. When you configure HTTPS, you must combine the intermediate certificates and server certificate into a complete certificate before you upload it.

> **Note** When you combine the certificates, make sure that the server certificate is followed by the intermediate certificate. In most cases, the CA provides the instructions when the CA issues a certificate. Pay attention to the instructions.

A chain of certificates that are issued by an intermediate CA:

```
-----BEGIN CERTIFICATE-----
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
-----END CERTIFICATE-----
```

The certificates in the chain must comply with the following rules:

- Empty lines are not allowed between certificates.
- Each certificate must be in PEM format.

RSA private key formats

A Rivest-Shamir-Adleman (RSA) private key must comply with the following rules:

- Run the `openssl genrsa -out privateKey.pem 2048` command to generate the RSA private key. `privateKey.pem` is the private key file.
- The private key must start with `-----BEGIN RSA PRIVATE KEY-----` and end with `-----END RSA PRIVATE KEY-----`.
- All lines except the last line must be 64 characters in length. The last line can be less than 64 characters in length.

If you do not generate the private key as instructed and the private key does not start with `-----BEGIN PRIVATE KEY-----` or end with `-----END PRIVATE KEY-----`, run the following command to convert the private key:

```
openssl rsa -in old_server_key.pem -out new_server_key.pem
```

Then, upload the `new_server_key.pem` file and the certificate together.

Convert certificate formats

HTTPS configuration supports only certificates that are in the PEM format. If your certificates are not in the PEM format, you must convert them to the PEM format. We recommend that you use OpenSSL to convert certificate formats. This section describes how to convert certificates to PEM:

- Convert a certificate from DER to PEM

  The DER format is typically used for Java.

  - Convert the certificate format:

    ```
    openssl x509 -inform der -in certificate.cer -out certificate.pem
    ```

  - Convert the private key format:

    ```
    openssl rsa -inform DER -outform pem -in privatekey.der -out privatekey.pem
    ```

- Convert a certificate from P7B to PEM

  The P7B format is typically used for Windows Server and Tomcat.

  - Convert the certificate format:

    ```
    openssl pkcs7 -print_certs -in incertificat.p7b -out outcertificate.cer
    ```

    Open the `outcertificat.cer` file. Then, upload the part that starts with `-----BEGIN CERTIFICATE-----` and ends with `-----END CERTIFICATE-----`.

  - A certificate in the P7B format does not include a private key. When you configure an SSL certificate in the DCDN console, specify the certificate information. You do not need to specify the private key.

- Convert a certificate from PFX to PEM

  The PFX format is typically used for Windows Server.

  - Convert the certificate format:

    ```
    openssl pkcs12 -in certname.pfx -nokeys -out cert.pem
    ```

  - Convert the private key format:

    ```
    openssl pkcs12 -in certname.pfx -nocerts -out key.pem -nodes
    ```

9.3. Configure an SSL certificate

Dynamic Route for CDN (DCDN) supports HTTPS secure acceleration. You can upload a custom certificate or add a purchased certificate from SSL Certificates Service to DCDN to enable HTTPS secure acceleration. HTTPS secure acceleration implements encryption for data in transit. This topic describes how to configure and renew an SSL certificate.

Prerequisites

-
- If you want to use a custom certificate, it must be in a valid format. For more information, see Certificate formats.

Context

DCDN supports only certificates in the `PEM` format. If a certificate is not in the `PEM` format, convert the certificate to the `PEM` format. For more information, see Convert certificate formats.

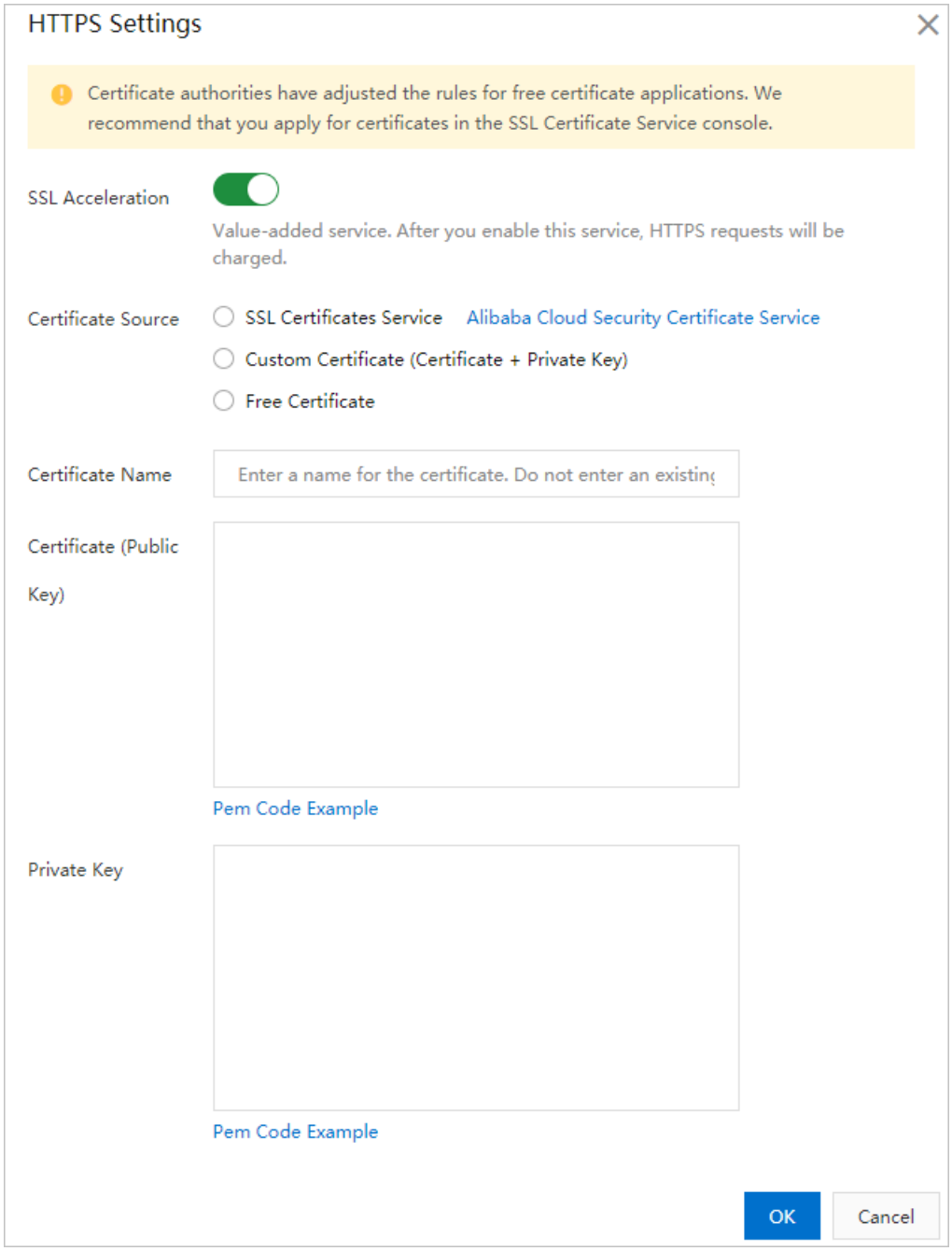
Note

Step 1: Configure or renew an SSL certificate

HTTPS secure acceleration is a value-added service. After you enable HTTPS, you are charged based on the number of HTTPS requests. You cannot use DCDN data transfer plans to offset the fees. For more information about HTTPS pricing, see Billing of requests and WebSocket.

1.
2.
3.
4.
5. In the **HTTPS Certificate** section, click **Modify**.
6. In the **HTTPS Settings** dialog box, turn on **SSL Acceleration**.

   After you turn on **SSL Acceleration**, the system prompts that you will be charged for using HTTPS. You can choose to enable or disable HTTPS based on your business requirements. For more information about HTTPS pricing, see Billing of requests and WebSocket.
7. On the page that appears, configure parameters.



Parameter	Description
Certificate Source	**Certificate Source** contains an Alibaba Cloud certificate, a custom certificate, and a free certificate. You can switch among these three types of certificates based on your business requirements. ○ ○ ○
Certificate Name	

Parameter	Description
Certificate (Public Key)	If you set the **Certificate Source** parameter to **Custom Certificate (Certificate + Private Key)**, you must set the **Certificate (Public Key)** and **Private Key** parameters. For more information, see **PEM Encoding Reference** below the **Certificate (Public Key)** and **Private Key** fields.
Private Key	

8. Click **OK**.

Step 3: Disable HTTPS secure acceleration

If you no longer require HTTPS secure acceleration, you can disable this feature in the DCDN console at any time. HTTPS secure acceleration is disabled immediately after you turn off the switch.

9.4. Enable HTTP/2

HTTP/2 is the latest version of HTTP. HTTP/2 enables more secure and efficient access to resources. This topic describes the concept and benefits of HTTP/2. This topic also describes how to enable HTTP/2.

Prerequisites

An HTTPS certificate is configured. For more information, see Configure an SSL certificate.

> **Note**
> - If you are configuring an HTTPS certificate for the first time, you must wait for the certificate to take effect before you enable HTTP/2.
> - If you disable HTTPS certificates after your enable HTTP/2, HTTP/2 is automatically disabled.

Context

HTTP/2, originally named HTTP 2.0, is the latest version of HTTP. It is supported by all major browsers such as Google Chrome, Internet Explorer 11, Safari, and Mozilla Firefox. HTTP/2 provides optimized performance and is compatible with HTTP/1.1 semantics. HTTP/2 is similar to SPDY but differs greatly from HTTP/1.1.

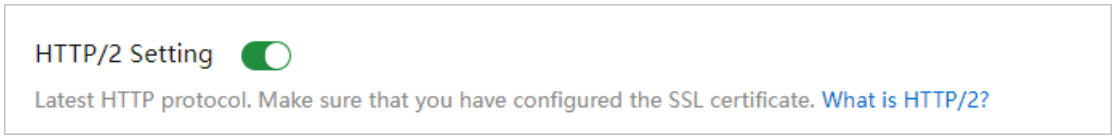
Benefits of HTTP/2:

- Binary encoding: Unlike HTTP 1.x that parses data into texts, HTTP/2 splits the data to be transmitted into messages and frames and encodes them into binary formats. Binary encoding makes HTTP/2 more scalable. For example, frames can be introduced to transmit data and instructions.
- Content security: HTTP/2 is designed based on HTTPS, protecting content security while maintaining network performance.
- Multiplexing: HTTP/2 allows multiplexing of multiple concurrent streams on a single connection. Specifically, you can initiate countless requests at the same time over one connection by using a browser, and the server returns the responses to these requests at the same time. In addition, you can set stream dependencies, which the client uses to inform the server of the importance of a given stream relative to other streams on the same connection, so that resources can be allocated appropriately.
- Header compression: HTTP headers carry large volumes of information, which is transmitted

repeatedly. HTTP/2 compresses HTTP headers into the HPACK format, allowing both ends of the communications to each cache a copy of the HTTP header indexes and hence transmit only index numbers for duplicate HTTP headers. This increases transmission speed and efficiency.

Procedure

1.
2.
3.
4.
5. In the **HTTP/2 Setting** section, turn on **HTTP/2**.



9.5. Configure OCSP stapling

Online Certificate Status Protocol (OCSP) stapling allows Dynamic Route for CDN (DCDN) nodes to cache the revocation status of SSL certificates and return the information to clients. Clients do not need to query the revocation status of SSL certificates from certificate authorities (CAs). This reduces the time that is required for the certificate validation process. This topic describes the OCSP stapling feature, the prerequisites for enabling OCSP stapling, and how to enable OCSP stapling.

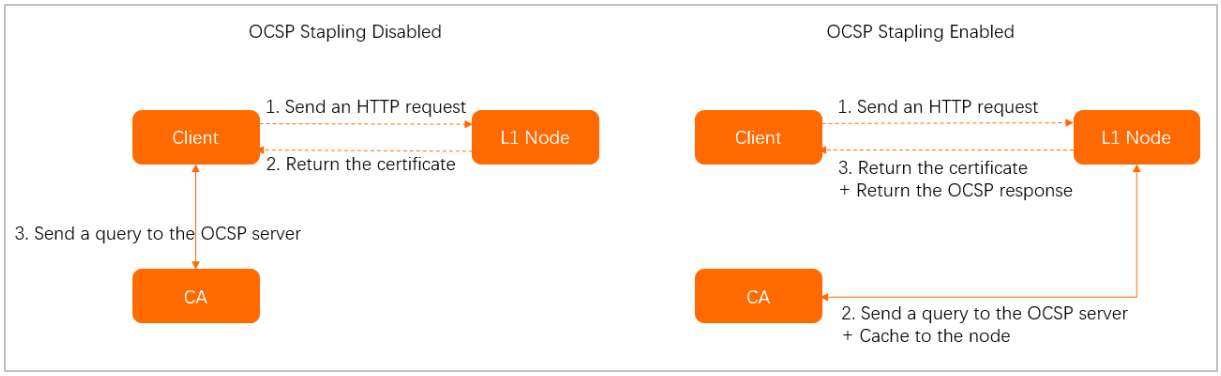
This topic consists of the following sections:

- Overview
- Prerequisites
- Procedure

Overview

The OCSP information is provided by CAs. Clients can use OCSP to check the revocation status of SSL certificates.

After OCSP stapling is enabled, the query process is performed by DCDN nodes. DCDN sends requests to retrieve OCSP information at a low frequency and caches the retrieved OCSP information on DCDN nodes. The default time-to-live (TTL) for cached OCSP information is 60 minutes. When a client sends a Transport Layer Security (TLS) handshake request to DCDN, DCDN returns the certificate and OCSP information to the client. The client can check the revocation status of the certificate without sending queries to the CA. This improves the TLS handshake efficiency and reduces the validation time.



> Notice
> - By default, OCSP stapling is disabled.
> - The default TTL of cached OCSP information is one hour. After the information expires, OCSP stapling does not take effect until the OCSP information is acquired again.
> - You can enable or disable OCSP stapling for accelerated domain names that have HTTPS secure acceleration enabled. If you delete the certificate settings, OCSP stapling is disabled.
> - The OCSP stapling process does not raise security risks because the OCSP information of digital certificates cannot be forged.

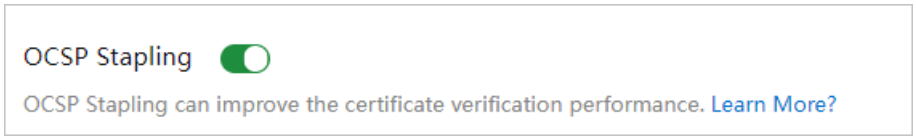
Prerequisites

Make sure that the following prerequisites are met before you configure OCSP stapling:

- An SSL certificate is configured. For more information, see Configure an SSL certificate.
- OCSP-specific extension fields are supported by clients. Otherwise, OCSP stapling cannot take effect.
- A medium or high number of queries per second (QPS) is maintained by your workloads. Otherwise, OCSP stapling cannot take effect.

Procedure

1.
2.
3. On the **Domain Names** page, find the domain name that you want to manage and click **Configure** in the Actions column.
4. In the left-side navigation pane of the domain name, click **HTTPS Settings**.
5. In the **OCSP Stapling** section, turn on **OCSP Stapling**.



9.6. Configure force redirect

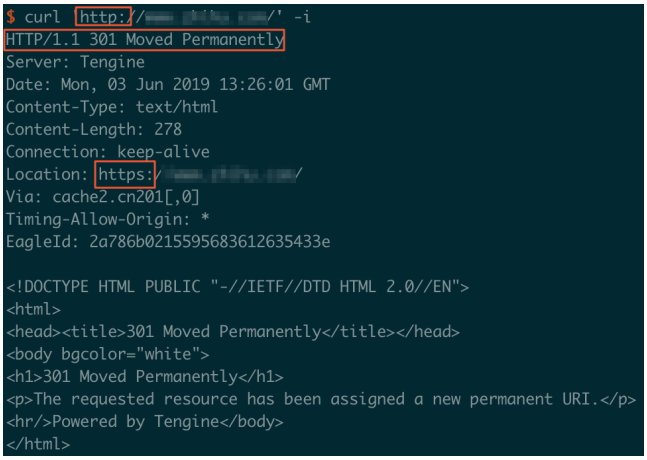
Prerequisites

An HTTPS certificate is configured. For more information, see Configure an SSL certificate.

Scenarios

HTTP and HTTPS force redirect features are suitable for the following scenarios:

- For accelerated domain names that have SSL certificates configured, you can enable 301 redirection to redirect HTTP requests between clients and Dynamic Route for CDN (DCDN) nodes to HTTPS. Compared with HTTP, HTTPS provides reinforced protection.



- For security-insensitive applications, you can enable 301 redirection to redirect HTTPS requests between clients and DCDN nodes to HTTP.

By default, the force redirect feature uses the HTTP 301 status code. You can change the HTTP status code to 308. To change the HTTP status code, submit a ticket.

HTTP status code	Description	Processing method	Use scenario
301	Moved Permanently	GET requests remain unchanged. Requests that use other request methods may be changed to GET.	Website refactoring.
308	Permanent Redirect	Both the request method and message body remain unchanged.	Website refactoring. This HTTP status code is suitable for requests that use request methods other than GET (with non-GET links/operations).

Billing rules

- If you set **Force Redirect** to **HTTPS to HTTP**, you are charged for HTTPS requests before redirection. HTTP requests that are redirected from HTTPS are free of charge.
- If you set **Force Redirect** to **HTTP to HTTPS**, you are charged for HTTPS requests that are redirected from HTTP. HTTP requests before redirection are free of charge.

Procedure

1.
2.
3.
4.
5. In the **Force Redirect** section, click **Modify**.
6. In the **Force Redirect** dialog box, select a **Redirect Type**.

Redirect type	Description
Default	Both HTTP and HTTPS requests are supported.
HTTPS to HTTP	Redirects client requests from HTTPS to HTTP.
HTTP to HTTPS	Redirects client requests from HTTP to HTTPS for secure data transmission.

7. Click **OK**.

9.7. Configure TLS version control

Alibaba Cloud Dynamic Route for CDN (DCDN) supports Transport Layer Security (TLS) version control. You can enable TLS versions for your domain names based on your business requirements. Early versions of TLS support browsers of earlier versions, but provide relatively low security. The latest versions of TLS provide enhanced security, but may not be compatible with browsers of earlier versions. This topic describes the concepts, use scenarios, and configuration method of TLS version control.

TLS versions

TLS is designed to ensure the security and integrity of data transmitted between two applications. A typical use case of TLS is HTTPS. HTTPS, also known as HTTP over TLS, is a secure version of HTTP. HTTPS runs below the top application layer (HTTP) and above the transport layer (TCP), and provides data encryption and decryption services.

Version	Description	Supported mainstream browser
TLSv1.0	TLS 1.0 was defined in RFC 2246 in 1999 as an update to SSL 3.0. TLS 1.0 is vulnerable to various attacks, such as BEAST and POODLE attacks. TLS 1.0 is no longer recommended for network protection due to the weak encryption performance. TLS 1.0 is not compliant with Payment Card Industry Data Security Standard (PCI DSS).	• Internet Explorer 6 and later • Google Chrome 1 and later • Firefox 2 and later
TLSv1.1	TLS 1.1 was defined in RFC 4346 in 2006 as an update to TLS 1.0. TLS 1.1 fixed some vulnerabilities in TLS 1.0.	• Internet Explorer 11 and later • Google Chrome 22 and later • Firefox 24 and later • Safari 7 and later
TLSv1.2	TLS 1.2 was defined in RFC 5246 in 2008 and is a widely used TLS version.	• Internet Explorer 11 and later • Google Chrome 30 and later • Firefox 27 and later • Safari 7 and later

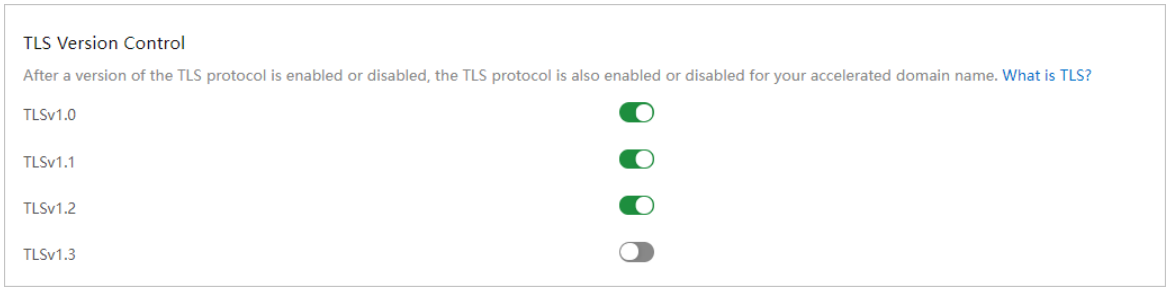
Version	Description	Supported mainstream browser
TLSv1.3	TLS 1.3 was defined in RFC 8446 in 2018 as the latest TLS version. TLS 1.3 supports the zero round trip time resumption (0-RTT) mode and allows you to establish faster connections. TLS 1.3 supports only key exchange algorithms of perfect forward secrecy to enhance security.	• Google Chrome 70 and later • Firefox 63 and later

Procedure

An SSL certificate is configured for the domain name. For more information, see Configure an SSL certificate.

> **Note** By default, TLS 1.0, TLS 1.1, and TLS 1.2 are enabled.

1.
2.
3.
4.
5. In the **TLS Version Control** section, enable or disable specific TLS versions based on your business requirements.



Recommended versions

Scenario	Recommended version
Require compatibility with browsers of earlier versions and security is not a priority	TLS 1.0, TLS 1.1, and TLS 1.2
Security is a priority and incompatibility with some browsers is acceptable	TLS 1.2
Early adopters	TLS 1.0, TLS 1.1, TLS 1.2, and TLS 1.3

9.8. Configure HSTS

This topic describes how to configure HTTP Strict Transport Security (HSTS). After HSTS is configured, clients such as browsers can establish only HTTPS connections to Dynamic Route for CDN (DCDN) nodes. HSTS protects requests from hijacking.

Prerequisites

An SSL certificate is configured for the domain name. For more information, see Configure an SSL certificate.

Context

HSTS is a policy mechanism that allows websites to accept only HTTPS connections. Websites can use HSTS to demand that clients such as browsers must use HTTPS. All HTTP requests and untrusted SSL certificates are rejected. HSTS prevents man-in-the-middle (MITM) attacks during the first visits from clients.

If HSTS is disabled and HTTPS is enabled on DCDN nodes, HTTP requests sent to the DCDN nodes are redirected to HTTPS based on the HTTP 301 or 302 status code when redirection from HTTP to HTTPS is enabled. The first HTTP request sent from a client to a DCDN node may be hijacked or tampered with. Hijacking and tampering raise security issues. If HSTS is enabled, clients can access the origin server only over HTTPS. This prevents requests from hijacking and tampering.

The HSTS response header is provided in the format of `Strict-Transport-Security:max-age=expireTime [;includeSubDomains] [;preload]`. The following table describes the parameters in the header.

Parameter	Description
max-age	The time-to-live (TTL) of the HSTS header. Unit: seconds.
includeSubDomains	Optional. If this parameter is set, HSTS is enabled for the domain name and all subdomains of the domain name.
preload	Optional. This parameter allows you to add the domain name to the HSTS preloaded list of the browser.

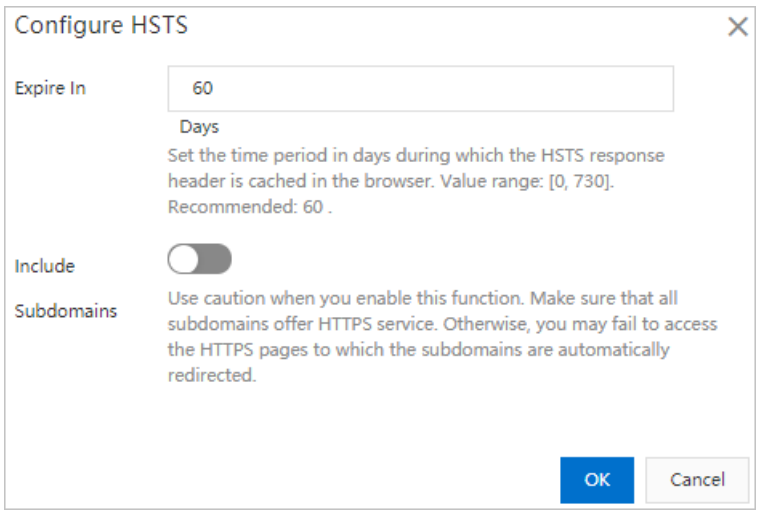
Limits

- Before HSTS takes effect, you can Configure force redirect to redirect the first HTTP request from a client to HTTPS by using 301 redirection.
- The HSTS response header applies to the responses to HTTPS requests but does not apply to the responses to HTTP requests.
- HSTS applies only to port 443.
- HSTS applies only to domain names. It does not apply to IP addresses.

Procedure

1.
2.
3.
4.
5. In the **HSTS** section, turn on **HSTS** and specify the **Expire In** and **Include Subdomains** parameters.
   - Expire In: specifies the TTL for the HSTS response header to be cached on the browser. You can specify a value between 0 and 730. We recommend that you set the value to 60. Unit: days.

- Include Subdomains: Proceed with caution. Make sure that HTTPS is enabled for all subdomains of the accelerated domain name. Otherwise, URLs to the subdomains become inaccessible after the requests are redirected to HTTPS.



6. Click **OK**.

9.9. Enable ShangMi for HTTPS

Alibaba Cloud provides the ShangMi (SM) for HTTPS feature to meet your security requirements. This topic describes how to enable ShangMi for HTTPS.

Prerequisites

- An SM certificate is purchased and deployed in the SSL Certificates Service console. For more information, see Step .

  > **Note** You must purchase an SM certificate in the SSL Certificates Service console. You cannot upload a custom SM certificate.

- An SSL certificate is configured for your domain name. For more information, see Configure an SSL certificate.

Background information

- ShangMi for HTTPS supports the SM2 algorithm and security protocols of the Chinese cryptographic standards. The SM2 algorithm is a public key cryptographic algorithm based on elliptic curves. You can use the Chinese cryptographic algorithm to establish an encrypted connection based on SSL and verify server identities. The browser that you use must support the Chinese cryptographic algorithm.
- Alibaba Cloud Dynamic Route for CDN (DCDN) provides more secure transmission over HTTPS based on the SM2 and SM3 algorithms. The SM2 algorithm is a public key cryptographic algorithm based on elliptic curves. The SM3 algorithm is a cryptographic hash algorithm.
- Cipher suites that are supported include ECC-SM2-WITH-SM4-SM3 and ECDHE-SM2-WITH-SM4-SM3. The cipher suites are used to verify whether the Chinese cryptographic algorithm is enabled.
- You can use the Chinese cryptographic algorithm for HTTPS only in Linux. If you use AliOS, you must deploy BabaSSL.

Procedure

1.
2. In the left-side navigation pane, choose **Content Delivery** > **Domain Names**.
3.
4. Find the domain name for which you want to enable ShangMi for HTTPS, click **HTTPS Settings** in the Actions column, and then click HTTPS Settings in the left-side navigation pane.
5. In the **ShangMi for HTTPS** section, turn on **ShangMi for HTTPS**.
6. (Optional) If the message **No SSL certificate is available** appears, click **Buy and Configure Certificate**.
   i. Log on to the SSL Certificates Service console to purchase a certificate.
   ii. Upload the certificate. For more information, see Upload a certificate.
7. If a certificate is available, select the certificate and click **OK** to enable ShangMi for HTTPS.
8. (Optional) If you want to disable the **ShangMi for HTTPS** feature, turn off **ShangMi for HTTPS** in the **ShangMi for HTTPS** section.

Related API operations

Operation	Description
SetDcdnDomainSMCertificate	Enables or disables an SM certificate for a domain name.
DescribeDcdnSMCertificateDetail	Queries the details about an SM certificate.
DescribeDcdnSMCertificateList	Queries the SM certificates of an accelerated domain name.

9.10. Enable authentication on client certificates

By default, SSL certificates can be used only for one-way authentication. These certificates are used by a client to verify the identity of a server. Alibaba Cloud Dynamic Route for CDN (DCDN) supports the authentication of client certificates. You can add a custom certificate authority (CA) certificate to verify the identity of the client for the server. This way, the client and the server can verify the identity of each other and communication between the website and the user is secured. This topic describes how to enable and configure the authentication on client certificates feature.

Prerequisites

- The SSL Certificate feature is enabled and configured. For more information, see Configure an SSL certificate.
- A **client CA certificate** is issued.

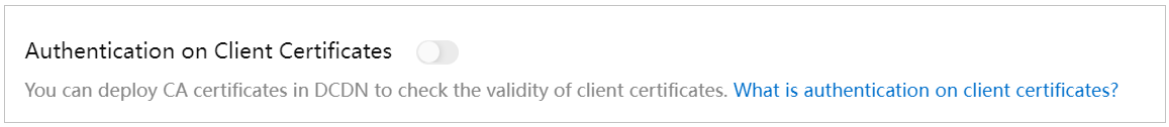
Procedure

1.
2. In the left-side navigation pane, click **Domain Names**.
3. On the **Domain Names** page, find the domain name that you want to manage and click **Configure**

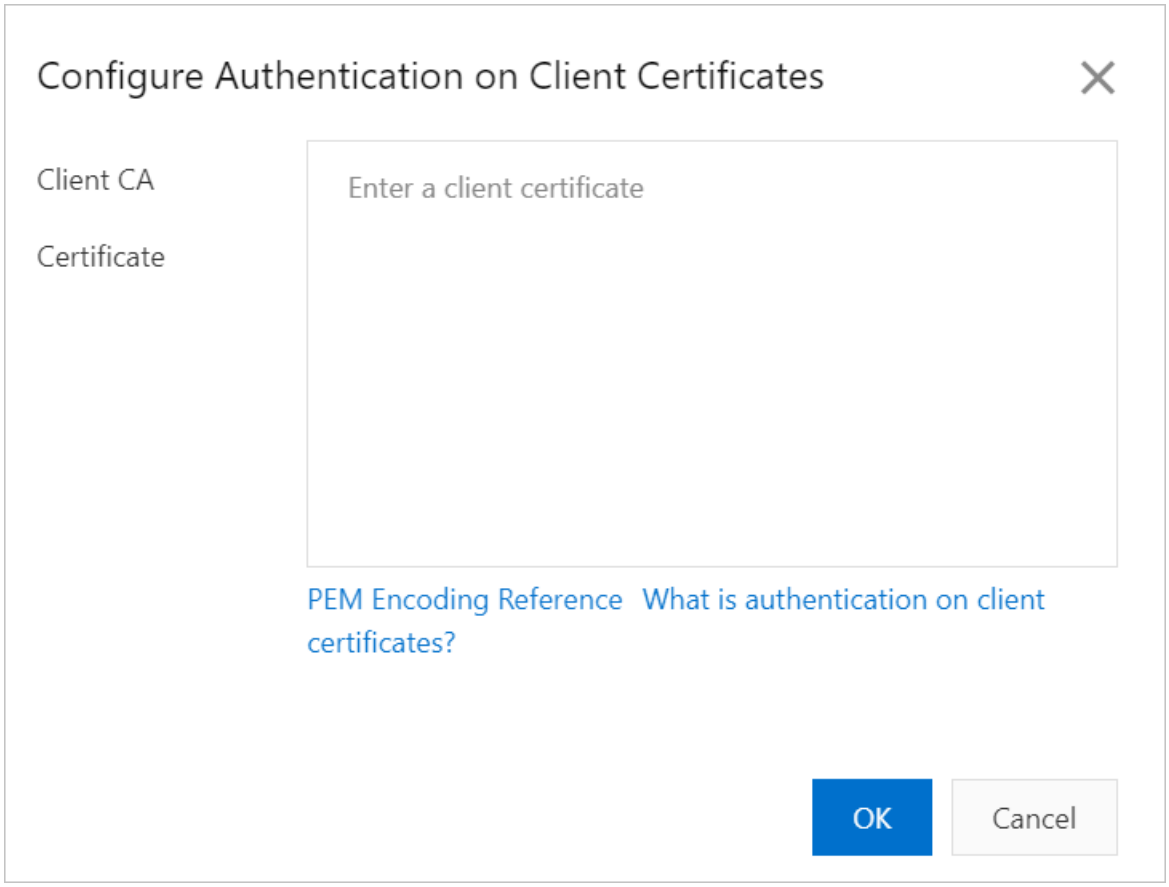
in the Actions column.

4. In the left-side navigation pane of the domain name, click **HTTPS Settings**.
5. Turn on **Authentication on Client Certificates**. In the Configure Authentication on Client Certificates dialog box, enter a custom **client CA certificate** in the Client CA Certificate field.

   Turn on **Authentication on Client Certificates**.

   Enter a custom **Client CA Certificate** in the Client CA Certificate field.

6. Click **OK**.
   After you enable **Authentication on Client Certificates**, DCDN checks whether the certificate of a client is valid when the client sends an HTTPS request. If the certificate of the client is valid, the request is allowed. Otherwise, the request is rejected.

10.Access Control

10.1. Overview

You can use Referer-based hotlink protection, URL signing, IP blacklists and whitelists, and User-Agent blacklists and whitelists to recognize and filter user requests. These features regulate access to Alibaba Cloud DCDN and improve service security.

You can use the following features to implement access control.

Feature	Description
Configure a referer whitelist or blacklist to enable hotlink protection	Allows you to configure a Referer whitelist or blacklist to recognize and filter user requests. This feature regulates access to Alibaba Cloud DCDN and protects your websites from hotlinks.
Configure URL authentication	Allows origin servers to authenticate requests based on signatures. URL signing supports custom signature strings and timestamps, and protects origin servers from unauthorized access. Compared with Referer-based hotlink protection, URL signing provides enhanced protection and is suitable for protecting security-sensitive files.
Configure an IP address blacklist or whitelist	Allows you to configure an IP whitelist or blacklist to recognize and filter user requests. This feature regulates access to Alibaba Cloud DCDN and prevents security issues such as IP theft and attacks.
Configure a User-Agent blacklist or whitelist	A user agent identifies a client. You can configure a User-Agent blacklist or whitelist to recognize and filter user requests. Only authorized clients have access to your resources.

10.2. Configure a referer whitelist or blacklist to enable hotlink protection

You can configure a Referer whitelist or blacklist to specify whether Referer headers with empty values are allowed to access your resources. You can use the Referer header to control access to resources and protect websites from unauthorized access.

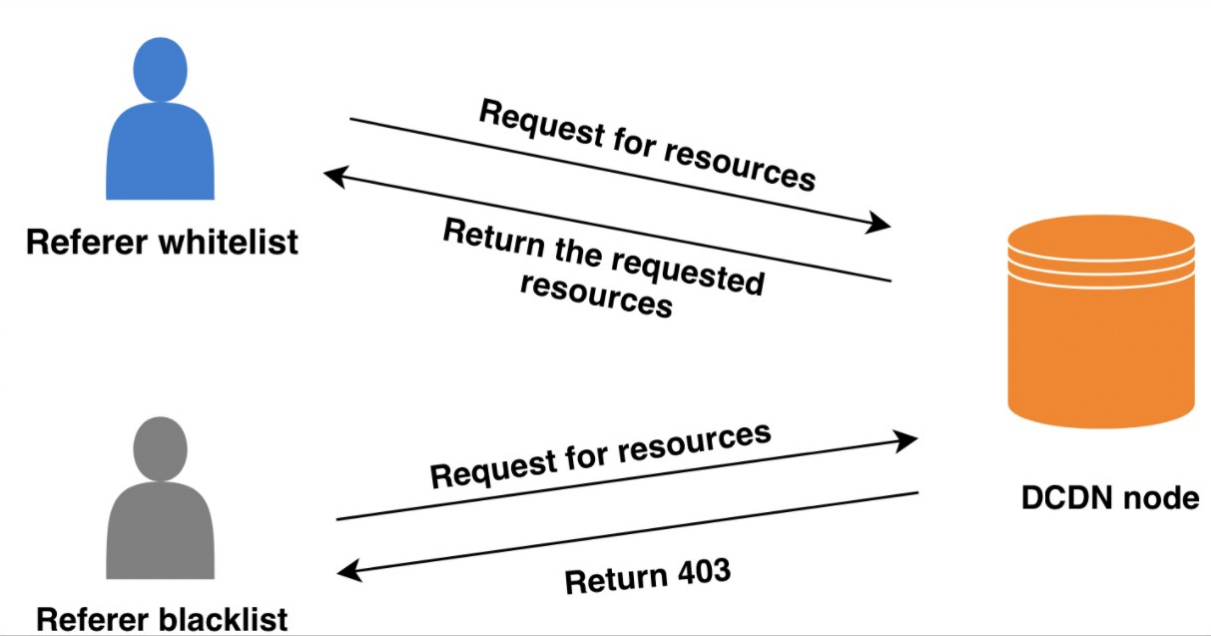
Context

Notice

- By default, hotlink protection is disabled.
- After you add a domain name to the referer whitelist or blacklist, the wildcard domain name that the domain name matches is automatically added to the whitelist or blacklist. For example, if you add `example.com` to the whitelist or blacklist, the domain name that takes effect is `*.example.com`. Hotlink protection takes effect on all domain names that match *.example.com.

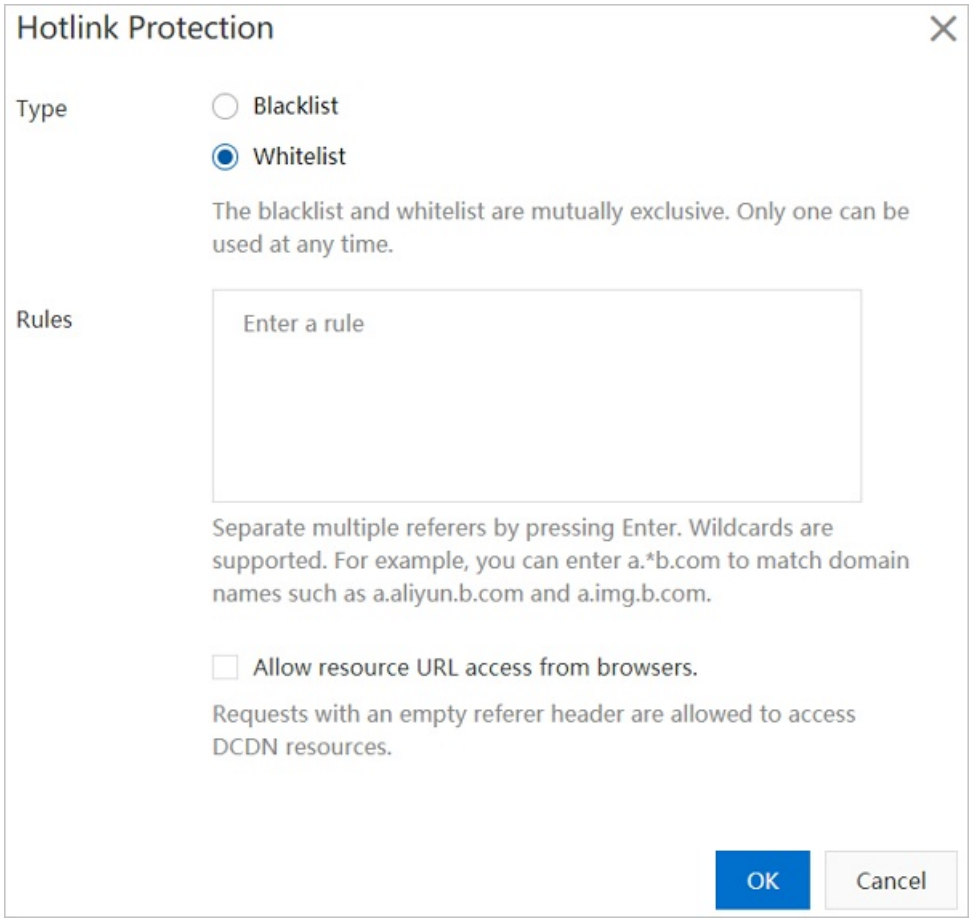
The Referer header is a component of the header section in HTTP requests and carries information about the source address, including the protocol, domain name, and query string. Referer is used to identify the source of a request.

You can configure a referer whitelist or blacklist to identify the sources of requests that are sent to Alibaba Cloud Dynamic Route for DCDN nodes, and determine whether to allow the requests to access your resources. If a request is allowed, DCDN returns the URL of the requested resource. If a request is not allowed, DCDN returns an HTTP 403 status code.



Procedure

1.
2.
3.
4.
5. On the **Hotlink Protection** tab, turn on **Hotlink Protection**.
6. Select **Blacklist** or **Whitelist** based on your business requirements.



Parameter	Description
Type	◦ Blacklist Requests from the domain names in the blacklist cannot access the current resource. ◦ Whitelist Only requests from the domain names in the whitelist are allowed to access the current resource. **Note** Blacklists and whitelists are mutually exclusive. You can configure only one of them.
Rules	◦ You can add multiple domain names to the Referer whitelist or blacklist. Separate domain names with carriage return characters. ◦ You can use an asterisk (*) to specify wildcard domain names. For example, if you specify `.*developer.aliyundoc.com`, `image.developer.aliyundoc.com` and `video.developer.aliyundoc.com` match the wildcard domain name.

Parameter	Description
Allow resource URL access from browsers.	If you select this check box, requests that have empty referer values or do not carry the referer field, such as requests sent from browsers, are allowed to access the requested resource regardless of the referer whitelist or blacklist.

7. Click **OK**.

10.3. URL authentication

10.3.1. Configure URL authentication

By default, content distributed by Dynamic Route for CDN (DCDN) is publicly available. Users can access the content by using URLs. If you want to prevent your resources from hotlinking and unauthorized access, you can use referer whitelist and blacklist, IP whitelist and blacklist, and URL authentication to regulate access control. URL authentication adds signature strings and timestamps to URLs to enhance access control. This topic describes how URL authentication works, how to enable or disable URL authentication, and how to verify the URL authentication settings.

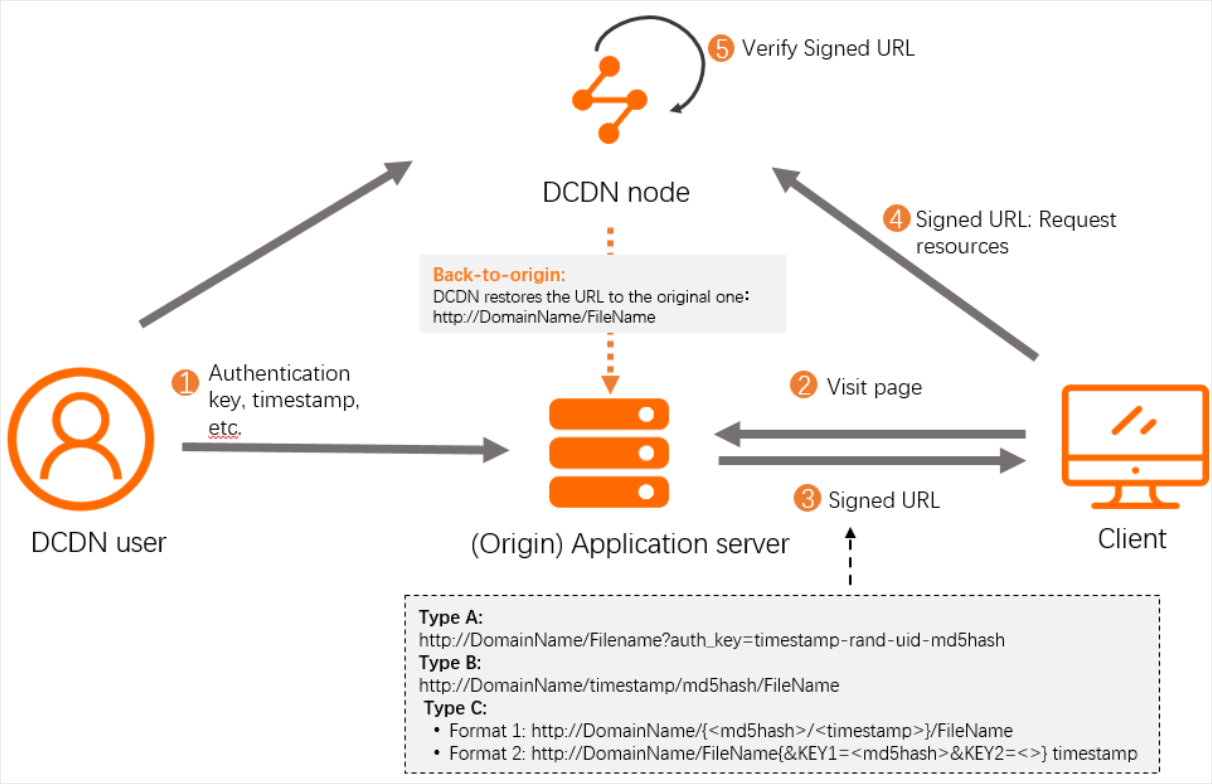
This topic consists of the following sections:

- How URL authentication works
- Configure and enable URL authentication
- Check the URL authentication result
- Disable URL authentication

How URL authentication works

DCDN nodes work with origin servers to implement URL authentication to protect resources on the origin servers in a more secure and reliable manner. URL authentication involves the following objects:

- Origin server: The origin server encrypts URLs based on the URL authentication rules, including authentication algorithms and cryptographic keys. Then, the origin server returns the encrypted URLs to clients.
- Client: The client initiates a request and sends the encrypted URL to DCDN nodes for authentication.
- DCDN nodes: The DCDN nodes verify the authentication information, including the signature and timestamp, in the encrypted URL.



1. You must configure URL authentication rules, including authentication algorithms and cryptographic keys, on your origin server.

   For example, `http://DomainName/timestamp/md5hash/FileName` is a URL encrypted by the origin server.

2. When a client attempts to access a URL, the origin server encrypts the URL based on the authentication rules, and then returns the encrypted URL to the client, as shown in Step 2 and Step 3 in the preceding figure.
3. The client uses the encrypted URL to request resources from DCDN nodes, as shown in Step 4 in the preceding figure.
4. The DCDN nodes check the authentication information, including the signature string and timestamp, in the encrypted URL and determine whether the request is valid, as shown in Step 5 in the preceding figure.
   - **If the request fails the authentication, DCDN nodes reject the request.**
   - **If the request passes the authentication, DCDN nodes respond to the request.**

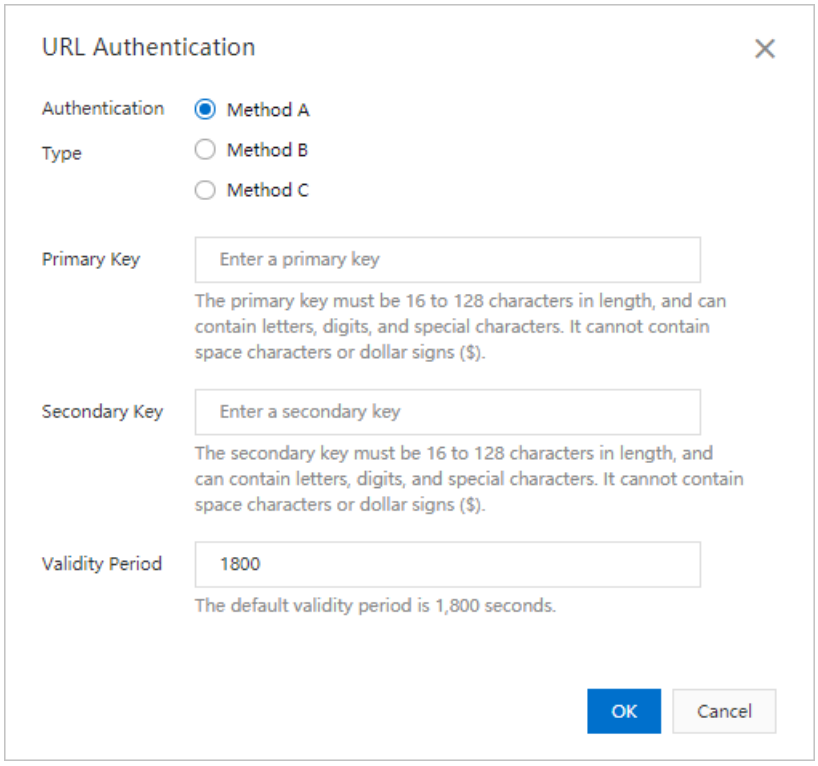
> **Note**
>
> - If the requested resource is not cached on DCDN nodes, the nodes remove the authentication parameters from the URL and restore the URL to the original version before the request is redirected to the origin server. For example, the URL is restored to `http://DomainName/FileName`. Then, the original URL is used to generate a cache key or initiate a back-to-origin request.
> - After a request passes the authentication, the special characters such as equal signs (`=`) and plus signs (`+`) in the URL are escaped.

Configure and enable URL authentication

Notice

- Before you enable URL authentication, make sure that you have configured URL authentication rules, including authentication algorithms and cryptographic keys, on the origin server.
- The authentication logic on DCDN nodes must be the same as that on the origin server.

1.
2.
3.
4.
5. Click the **URL Authentication** tab.
6. Turn on **URL Authentication Setting**.
7. In the **URL Authentication** dialog box, configure URL authentication parameters.



Parameter	Description

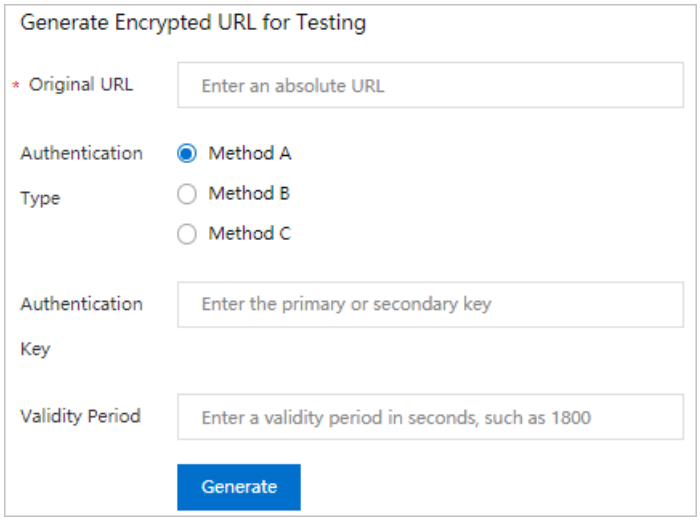
Parameter	Description
Authentication Type	Alibaba Cloud DCDN supports three URL authentication methods. You can select an authentication method based on your business requirements to protect resources on your origin server. Supported authentication methods are: ○ Type A signing ○ Authentication type B ○ Authentication type C **Note** If a URL authentication error occurs, a 403 error is returned. ○ Invalid MD5 values Example: `X-Tengine-Error:denied by req auth: invalid md5hash=de7bfdc915ced05e17380a149bd760be` ○ Invalid timestamps Example: `X-Tengine-Error:denied by req auth: expired timestamp=1439469547`
Primary Key	Specify the primary key for the selected authentication method.
Secondary Key	Specify the secondary key for the selected authentication method.
Validity Period	Specify a validity period for encrypted URLs. Users can access DCDN nodes before the encrypted URLs expire. The expiration time of an encrypted URL is determined by the timestamp value and the specified validity period. ○ Unit: seconds. ○ Valid values: 1 to 31536000. ○ Default value: 1800, which equals 30 minutes. ○ For example, the timestamp of an encrypted URL is 2020-08-15 15:00:00 (UTC+8), and the value of validity period is 1800. In this case, the encrypted URL remains valid until 15:30:00 on August 15, 2020 (UTC+8).

8. Click **OK**.

Check the URL authentication result

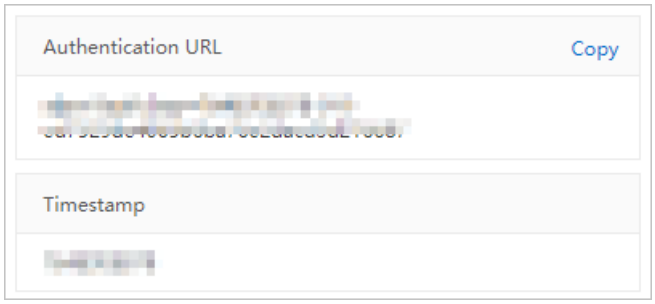
To ensure that the authentication logic is correctly implemented, we recommend that you run a test in the DCDN console to verify whether the encrypted URLs are correct.

1. In the **Generate Signed URL** section, configure **Original URL** and other parameters.



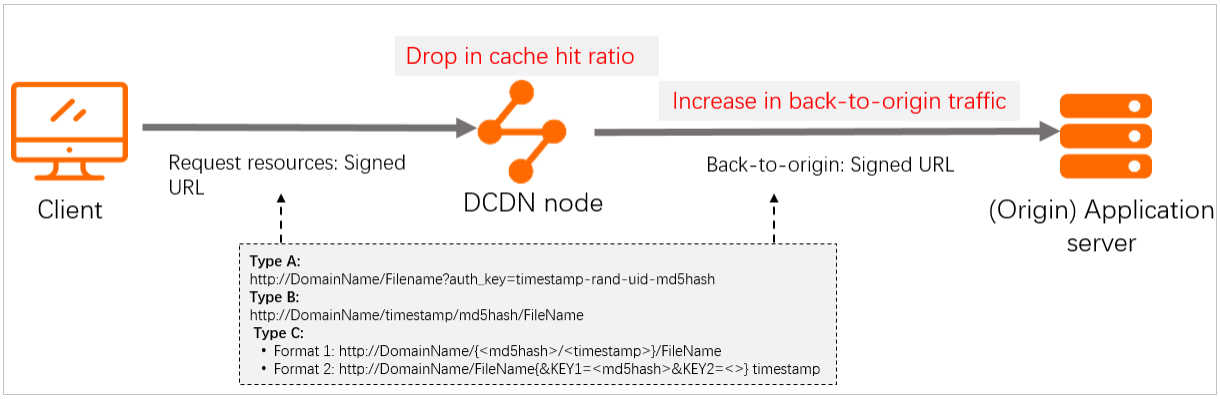
Parameter	Description
Original URL	Enter a complete URL, for example, `https://www.aliyun.com`.
Type	Select the URL authentication method that you specified in Configure and enable URL authentication.
Cryptographic Key	Configure **Primary Key** or **Secondary Key** based on the key that you specified in Configure and enable URL authentication.
Validity Period	Enter the validity period of the encrypted URL that you specified in Configure and enable URL authentication.

2. Click **Generate** to obtain the **Authentication URL** and **Timestamp**.



Disable URL authentication

Notice If URL authentication is disabled on DCDN nodes, but user requests still carry authentication parameters, DCDN nodes fail to remove the authentication parameters. In this case, the requests cannot hit cache on DCDN nodes and are redirected to the origin server. This increases network traffic on the origin server and data transfer fees. If you want to disable URL authentication, make sure that URL authentication is disabled on both the origin server and DCDN nodes.



1. Log on to the DCDN console, navigate to the URL Authentication Setting section, and then turn off **Modify**.
2. On the origin server, delete the URL authentication settings.

10.3.2. Type A signing

Dynamic Route for CDN (DCDN) provides the URL signing feature to protect origin servers from unauthorized access and downloads. The URL signing feature supports three signing types. This topic describes how type A signing works.

How it works

URLs are signed in the following format:

```
http://DomainName/Filename?auth_key=timestamp-rand-uid-md5hash
```

The following table describes the fields in a signed URL.

Field	Description
DomainName	The accelerated domain name.
Filename	The actual URL that points to the requested resource on the origin server. The Filename field must start with a forward slash (/).
auth_key	The cryptographic key that you have set.
timestamp	The time when a URL expires. The value is a 10-digit positive integer. It specifies the number of seconds that have elapsed since 00:00:00 (UTC+8) on January 1, 1970 and the time-to-live (TTL) value of the cryptographic key. The TTL value of the cryptographic key is set on user clients. If the TTL value is set to 1,800 seconds by a user when they initiate a request, the URL of the request expires 1,800 seconds after the client connects to the CDN node. For example, if the connection between the client and origin server is established at 15:00:00 (UTC+8) on August 15, 2020 (2020-08-15 15:00:00), the URL of the request expires at 15:30:00 (UTC+8) on August 15 (2020-08-15 15:30:00).
rand	A random number. The number must not contain hyphens (-). Example: 477b3bbc253f467b8def6711128c7bec. We recommend that you use a universally unique identifier (UUID).

Field	Description
uid	The user ID. Set this field to 0.
md5hash	The string that is calculated by using the MD5 algorithm. It must be 32 characters in length, and can contain digits and lowercase letters.

When a DCDN node receives a request, it determines whether the `timestamp` in the request is earlier than the current time.

- If the timestamp is earlier than the current time, the DCDN node determines that the URL expires and returns a 403 error.
- If the timestamp is later than the current time, the DCDN node generates a string in the same format as the `sstring` string. It then uses the MD5 algorithm to calculate the `HashValue`, and compares it with the `md5hash` value in the request.
  - If they are the same, the request passes the authentication. The DCDN node returns the requested resource.
  - If they are different, the authentication fails. The CDN node returns a 403 error.

  The `HashValue` is calculated based on the following string:

```
sstring = "URI-Timestamp-rand-uid-PrivateKey". The URI specifies the address that points to the requested resource. It does not contain parameters such as /Filename.
HashValue = md5sum(sstring)
```

Example

The following example shows how to implement type A signing.

1. A user wants to retrieve a resource by using `req_auth`.

   ```
   http:// cdn.example.com/video/standard/1K.html
   ```

2. The cryptographic key is aliyuncdnexp1234.
3. The expiration time of the authentication configuration file is 00:00:00 (UTC+8) on October 10, 2015 (2015-10-10 00:00:00). Therefore, the validity period is 1,444,406,400 seconds.
4. The DCDN node generates a signature string to calculate the `HashValue`.

   ```
   /video/standard/1K.html-1444435200-0-0-aliyuncdnexp1234
   ```

5. The DCDN node calculates `Hashvalue` based on the signature string.

   ```
   HashValue = md5sum("/video/standard/1K.html-1444435200-0-0-aliyuncdnexp1234") = 80cd3862d699b7118eed99103f2a3a4f
   ```

6. Sign the request URL.

   ```
   http://cdn.example.com/video/standard/1K.html?auth_key=1444435200-0-0-80cd3862d699b7118eed99103f2a3a4f
   ```

If the `HashValue` calculated by the CDN node is the same as the `md5hash` value contained in the request (both are *80cd3862d699b7118eed99103f2a3a4f* in this example), the request passes the authentication. Otherwise, the request fails the authentication.

10.3.3. Authentication type B

The URL authentication feature protects resources on origin servers from unauthorized access and downloads. Dynamic Route for CDN (DCDN) provides you with three authentication types. This topic describes how authentication type B works and provides an example.

How it works

A request URL is encrypted in the following format:

```
http://DomainName/timestamp/md5hash/FileName
```

If the request passes authentication, the actual URL used to access the origin server is in the following format:

```
http://DomainName/FileName
```

The following table describes the fields in an encrypted URL.

Field	Description
DomainName	The domain name of the DCDN node.
timestamp	The time when the URL expires. The time is included in the URL and is used to calculate the `md5hash` value. The time follows the YYYYMMDDHHMM format. The time-to-live (TTL) value of a URL is 1,800 seconds. For example, if you set the access time to 15:00:00 (UTC+8) on August 15, 2020 (2020-08-15 15:00:00), the request URL expires at 15:30:00 (UTC+8) on August 15, 2020 (2020-08-15 15:30:00).
md5hash	The string that is calculated by using the MD5 algorithm. The string is 32 characters in length, and can contain digits and lowercase letters.
FileName	The actual URL that points to the requested resource on the origin server. The Filename field must start with a forward slash (/).

Example

The following example shows how to implement type-B authentication.

1. Retrieve the following object from the origin server.

   ```
   http://cdn.example.com/4/44/44c0909bcfc20a01afaf256ca99a8b8b.mp3
   ```

2. Set the key to aliyuncdnexp1234.
3. Set the time when the origin server is accessed to 201508150800.
4. The DCDN node creates a signature string to calculate `Hashvalue`.

   ```
   aliyuncdnexp1234201508150800/4/44/44c0909bcfc20a01afaf256ca99a8b8b.mp3
   ```

5. The DCDN node calculates the `md5hash` value based on the signature string `Hashvalue`.

```
md5hash = md5sum("aliyuncdnexp1234201508150800/4/44/44c0909bcfc20a01afaf256ca99a8b8b.mp3") = 9044548ef1527deadafa49a890a377f0
```

6. Encrypt the request URL.

```
http://cdn.example.com/201508150800/9044548ef1527deadafa49a890a377f0/4/44/44c0909bcfc20a01afaf256ca99a8b8b.mp3
```

If the value of `md5hash` calculated by the DCDN node is the same as the value of `md5hash` contained in the request (both are *9044548ef1527deadafa49a890a377f0*), then URL authentication succeeds. Otherwise, URL authentication fails.

10.3.4. Authentication type C

The URL authentication feature protects resources on origin servers from unauthorized access and downloads. Dynamic Route for CDN (DCDN) provides you with three authentication types. This topic describes how authentication type C works and provides an example.

How it works

A URL is encrypted in one of the following formats:

- Format 1

```
http://DomainName/{<md5hash>/<timestamp>}/FileName
```

- Format 2

```
http://DomainName/FileName{&KEY1=<md5hash>&KEY2=<timestamp>}
```

> **Note** The content enclosed by braces (`{}`) indicates the encrypted information that is added based on the standard URL format.

The following table describes the fields in an encrypted URL.

Field	Description
DomainName	The domain name of the DCDN node.
FileName	The actual URL that points to the requested resource on the origin server. The Filename field must start with a forward slash (/).
timestamp	The time when the origin server is accessed. The time must be in UNIX time. The value of the field is an unencrypted plaintext string. It is 10 digits in length. The value indicates the number of seconds that have elapsed since January 1, 1970. The number is a hexadecimal value.
md5hash	The string that is calculated by using the MD5 algorithm. The string 32 characters in length. It can contain digits and lowercase letters.

Example

The following example shows how to implement type-C authentication.

- Set the value of the PrivateKey field to `aliyuncdnexp1234`.
- Set the value of the FileName field to `/lesson-01.mp4`.
- Set the value of the timestamp field to `55CE8100`.
- Calculate the value of md5hash.

```
md5hash = md5sum(aliyuncdnexp1234/lesson-01.mp455CE8100) = a37fa50a5fb8f71214b1e7c95ec7****
```

- The following encrypted URLs may be generated:
  - Format 1:

```
http://example.aliyundoc.com/a37fa50a5fb8f71214b1e7c95ec7****/55CE8100/lesson-01.mp4
```

  - Format 2:

```
http://example.aliyundoc.com/lesson-01.mp4?KEY1=a37fa50a5fb8f71214b1e7c95ec7****&KEY2=55CE8100
```

When a client uses the encrypted URL to access a DCDN node, the DCDN node extracts encrypted string 1 and obtains `FileName` and access time from the original URL. The DCDN node performs the following steps to validate the request based on the defined business logic:

1. The CDN node uses the `Filename`, access time, and `PrivateKey` of the original URL to perform MD5 encryption. Encrypted string 2 is generated.
2. The DCDN node compares string 1 and string 2. If the two strings are different, the request is rejected.
3. The DCDN node checks whether the difference between its current time and the time in the original URL exceeds the time-to-live (TTL) value. By default, the value of TTL is 1,800 seconds.
   - If the time difference is smaller than the TTL value, the DCDN node returns a successful response.
   - If the time difference is greater than the TTL value, the DCDN node rejects the request and returns a 403 error.

> **Note** Assume that the TTL of 1,800 seconds is used for a request. If the difference between the time you access the origin server and the preset access time in the URL is greater than 1,800 seconds, the request fails authentication. For example, if you set the access time to 15:00:00 (UTC+8) on August 15, 2020 (2020-08-15 15:00:00), the URL expires at 15:30:00 (UTC+8) on August 15, 2020 (2020-08-15 15:30:00).

10.4. Configure an IP address blacklist or whitelist

An IP blacklist or whitelist filters user requests and blocks or allows requests from specified IP addresses. IP lists can protect origin servers from IP theft and attacks. This topic describes how to configure an IP blacklist or whitelist.

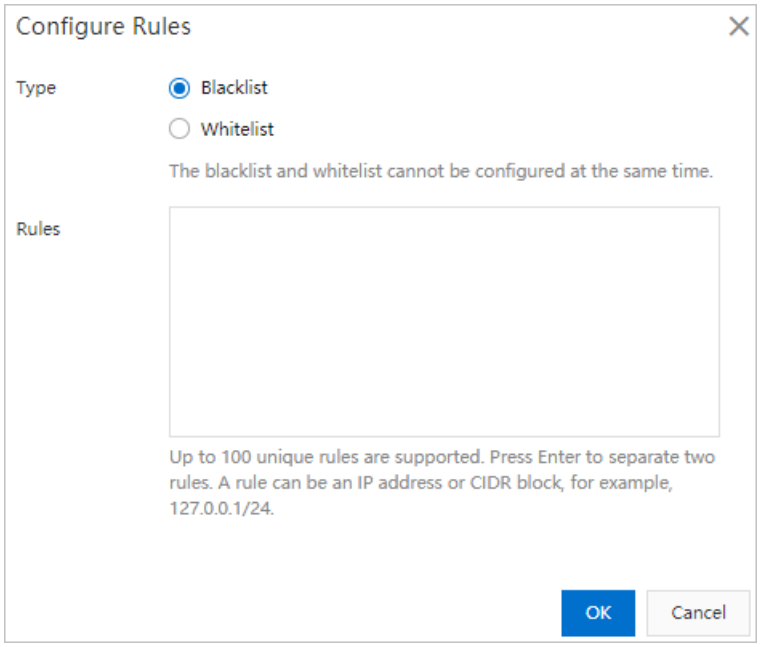
Usage notes

- By default, the IP list feature is disabled. The IP blacklist and whitelist are mutually exclusive. You can configure only one of them.

- If an IP address is added to the blacklist, requests from the IP address can still be sent to DCDN edge nodes. However, the DCDN edge nodes will reject the request and return a 403 error. Requests sent from IP addresses that are on the blacklist are recorded in the DCDN logs.
- The IP blacklist and whitelist identify IP addresses based on Layer 7 HTTP IP recognition techniques. Network traffic may be generated when DCDN edge nodes blocks requests. If clients access DCDN edge nodes over HTTPS, HTTPS request fees are incurred due to resources consumed for processing requests on DCDN edge nodes.

Procedure

1.
2.
3.
4.
5. On the right side of the page that appears, click the **IP Blacklist/Whitelist** tab.
6. Turn on **IP Blacklist/Whitelist**, and configure an IP address **Blacklist** or **Whitelist** as prompted.



Parameter	Description
Type	The following types of IP list are supported: ○ **Blacklist** Requests from IP addresses on the blacklist are blocked. ○ **Whitelist** Only requests from IP addresses on the whitelist are allowed to access resources on DCDN edge nodes.

Parameter	Description
Rules	Enter CIDR blocks such as 192.0.2.1/24 or IP addresses such as 192.168.0.1. Make sure that the CIDR blocks are not duplicate. Both IPv4 and IPv6 addresses are supported. You can add a maximum of 100 IP addresses to the whitelist or blacklist. Separate IP addresses with carriage return characters. IPv6: Both the blacklist and whitelist support IPv6 addresses. The letters in IPv6 addresses must be in uppercase, for example, 2001:DB8:0:23:8:800:200C:**** or 2001:0DB8:0000:0023:0008:0800:200C:****. The notation of an IPv6 address must not be shortened. For example, 2001:0DB8::0008:0800:200C:**** is invalid.

7. Click **OK**.

Configuration examples

- **Whitelist**

  CIDR clock: 192.0.2.1/24

  Expected result: only IP addresses that range from 192.0.2.1 to 192.0.2.254 (192.0.2.1 and 192.0.2.254 included) can access the resources of the accelerated domain name.

- **Blacklist**

  IP address: 192.168.0.1

  Expected result: The IP address 192.168.0.1 is not allowed to access the resources of the accelerated domain name.

10.5. Configure a User-Agent blacklist or whitelist

You can configure a User-Agent blacklist or whitelist to authenticate and filter visitors. This can restrict access to Dynamic Route for CDN (DCDN) resources and improve DCDN security. This topic describes how to configure a User-Agent blacklist or whitelist.

Context

If you need to implement access control based on the User-Agent field, you must configure a User-Agent blacklist or whitelist to filter requests.

- User-Agent blacklist: The User-Agent fields in the blacklist cannot be used to access resources.

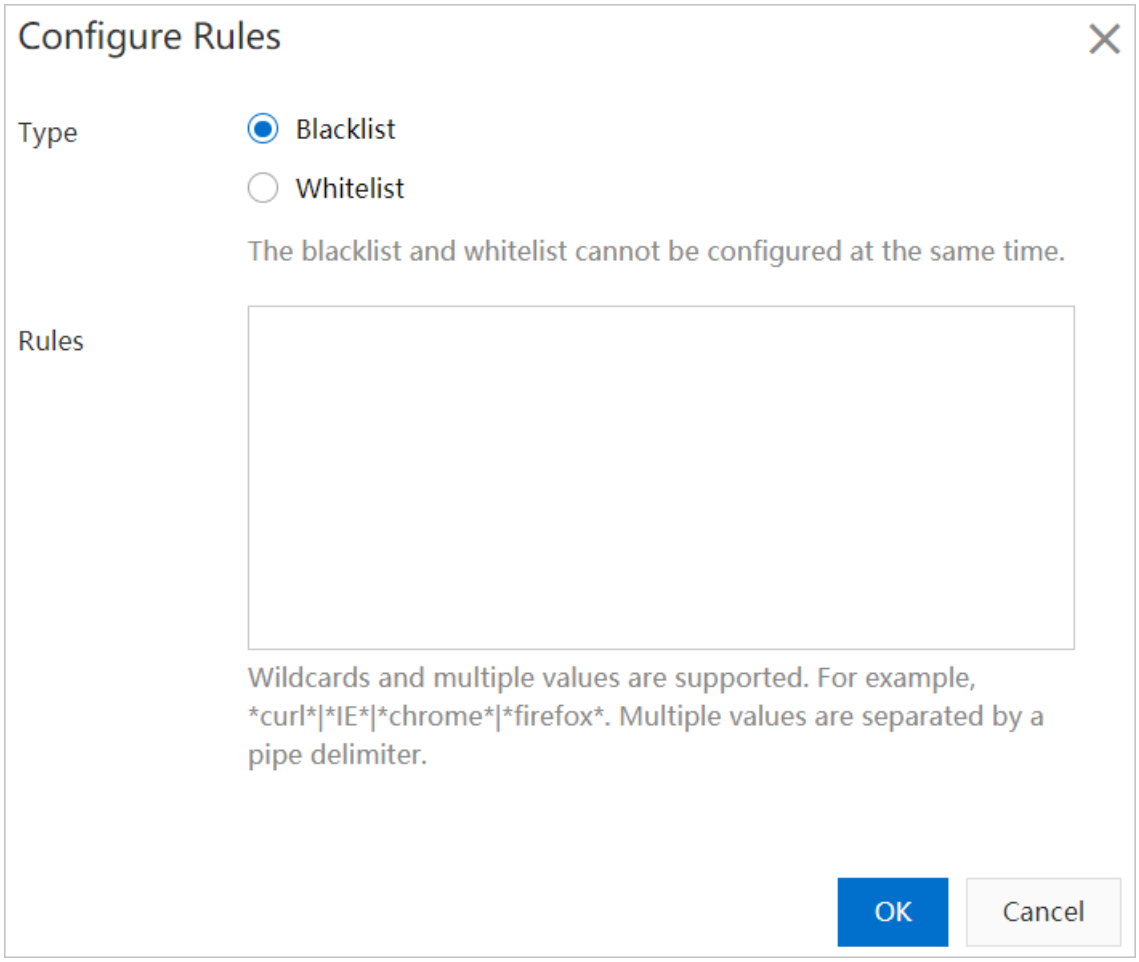
  If your User-Agent field is added to the blacklist, a request with the User-Agent field can be sent to a DCDN node. However, the DCDN node rejects the request and returns a 403 error. The requests that contain the User-Agent fields in the blacklist are recorded in DCDN logs.

- User-Agent whitelist: Only User-Agent fields in the whitelist can be used to access resources.

Procedure

1.
2.
3.

4.
5. On the right side of the page that appears, click the **User-Agent Black/White List** tab.
6. Turn on **User-Agent Black/White List**, and configure a User-Agent **Blacklist** or **Whitelist** as prompted.



Parameter	Description
Type	The following two types are supported: ○ **Blacklist** The User-Agent fields in the blacklist cannot be used to access the current resources. ○ **Whitelist** Only User-Agent fields in the whitelist can be used to access the current resources. Blacklists and whitelists are mutually exclusive. The most recent configuration takes effect.
Rules	When you configure the User-Agent fields, separate values with vertical bars (|). The User-Agent fields can contain wildcards (*), such as `*curl*|*IE*|*chrome*|*firefox*`

7. Click **OK**.

11.Performance Optimization

11.1. Overview

Dynamic Route for CDN (DCDN) provides multiple optimization features. These features help to reduce the size of the content that you want to access, accelerate content delivery, and improve the readability of the requested web pages.

DCDN supports the following optimization features.

Feature	Description
Configure HTML optimization	Enables DCDN to automatically remove redundant comments and duplicate spaces from all HTML pages. This helps to reduce file size and improve page readability.
Configure intelligent compression	Enables DCDN to compress static files by using Gzip. This helps to reduce the size of the transmitted files and accelerate content delivery.
Configure video seeking	Allows you to seek a specified position when you play video and audio content without compromising the playback quality.
Ignore parameters	Enables DCDN to ignore the parameters following a question mark (`?`) in the URL of a received request when it retrieves and caches the requested resource from the origin server.

11.2. Configure HTML optimization

Dynamic Route for CDN (DCDN) supports the HTML optimization feature. This feature enables DCDN to automatically remove redundant content from web pages, such as comments and additional whitespace characters in HTML pages and CSS or JavaScript code. This reduces file sizes, accelerates content delivery, and improves website readability.

Limits

- If MD5 validation is configured for a file on the origin server, do not enable the HTML optimization feature.

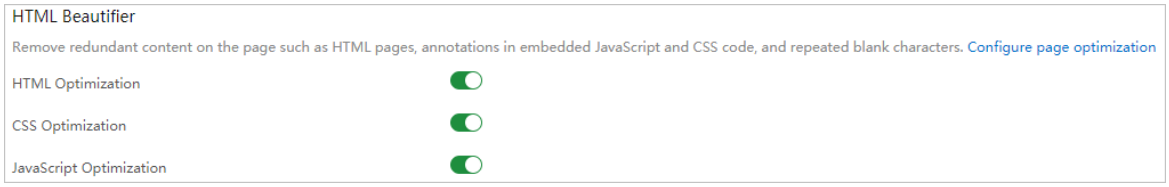
  The HTML optimization feature changes the MD5 value of a file. After the file is optimized, the MD5 value of the file is no longer the same as the original file stored on the origin server.
- If the origin server has Gzip or Brotli compression enabled, HTML optimization does not take effect. DCDN directly returns compressed files to clients.
- If both HTML optimization and Intelligent or Brotli compression are enabled, HTML optimization does not take effect. DCDN only compresses files.

Procedure

1.
2.
3.
4.

5. In the **HTML Beautifier** section, you can turn on **HTML Optimization**, **CSS Optimization**, or **JavaScript Optimization**.

> Note



Related API operations

BatchSetDcdnDomainConfigs

11.3. Configure intelligent compression

After you enable the intelligent compression feature, Alibaba Cloud Dynamic Route for CDN (DCDN) nodes compress resources before the resources are returned to clients. This reduces file sizes, accelerates file distribution, and reduces bandwidth consumption.

Context

- Both Gzip compression and Brotli compression are supported. Intelligent compression uses Gzip to compress files. For more information about Brotli compression, see Configure Brotli compression.
- Intelligent compression compresses files that are larger than 1 KB. Files smaller than 1 KB are not compressed.
- Intelligent compression supports the following formats: text/xml, text/plain, text/css, application/javascript, application/x-javascript, application/rss+xml, text/javascript, image/tiff, image/svg+xml, application/json, and application/xml.
- If a request carries the `Accept-Encoding: gzip` request header, the client expects the requested resources to be Gzip-compressed.
- If a response from the origin server carries the `Content-Encoding: gzip` response header, the resources returned to the client are Gzip-compressed.

Usage notes

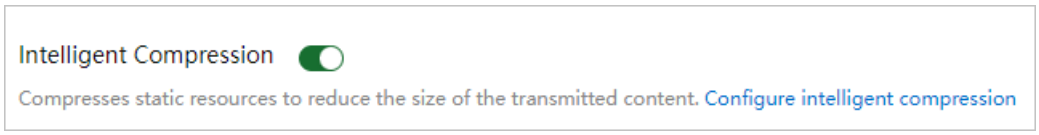
- Intelligent compression (Gzip compression) is compatible with all browsers. Brotli compression is not compatible with outdated browsers. You can query whether a browser supports Brotli compression as needed.
- When DCDN compresses static files, the MD5 values of the files are changed. If the origin server has MD5 verification enabled, disable intelligent compression and Brotli compression.
- If compression is enabled on the origin server and the response carries `content_encoding`, compression on DCDN nodes does not take effect.
- If both Brotli compression and intelligent compression are enabled, and the `Accept-Encoding` request header carries both `br` and `gzip`, only Brotli compression takes effect.
- If HTML optimization and file or Brotli compression are enabled, HTML optimization does not take effect. DCDN only compresses files.

- Common types of image files such as PNG, JPG, and JPEG and video files such as MP4, AVI, and WMV have already been compressed. Intelligent compression and Brotli compression do not take effect for these files. We recommend that you disable intelligent compression and Brotli compression. If you want to further reduce image file sizes, you can use the image editing feature. If you want to further reduce the sizes of video files, you can use the video transcoding feature.

Procedure

1.
2.
3.
4.
5. In the **Intelligent Compression** section, turn on **Intelligent Compression**.

   After intelligent compression is enabled, you can compare the file types before and after intelligent compression is enabled. If the file extension is .gzip, the file is compressed.



Related API operations

BatchSetDcdnDomainConfigs

11.4. Configure Brotli compression

To compress static text files, you can enable the Brotli compression feature to reduce the size of the transmitted content and accelerate delivery. This topic describes how to enable the Brotli compression feature.

Context

Brotli is a new open source compression algorithm. After you enable Brotli compression, Dynamic Content for CDN (DCDN) nodes compress text files such as HTML, JS, and CSS files when DCDN returns the requested resources. Brotli compression is 15% to 25% more efficient than gzip compression.

- If a request contains the `Accept-Encoding: br` header, this indicates that the requested resources are required to be compressed by using Brotli.
- If a response from DCDN includes the `Content-Encoding: br` header, this indicates that Brotli is used to compress the requested resources.

> **Notice**
> - If both Brotli compression and gzip compression are enabled, and the `Accept-Encoding` request header contains `br` and `gzip`, Brotli compression prevails over gzip compression.
> - If compression is enabled on an origin server and the response contains the `Content-Encoding` header, Brotli compression is not supported.

Procedure

1.
2.
3.
4. On the details page of the specified domain name, click **Optimization** in the left-side navigation pane.
5. In the **Brotli Compression** section, turn on **Brotli Compression**.

11.5. Configure image editing

11.5.1. Image editing and its benefits

Dynamic Route for CDN (DCDN) allows you to use image editing to edit and distribute images on edge nodes. This relieves pressure on origin servers, reduces the number of requests that are redirected to origin servers, and reduces the amount of network traffic generated during the back-to-origin process. You can use image editing to perform operations such as adding watermarks to images, resizing images, and cropping images based on your business requirements. The image editing feature of Alibaba Cloud CDN, DCDN, and Object Storage Service (OSS) are independent of each other and cannot be used together.

> **Note**
> - Before you use image editing, you must to activate this feature.
> - Image editing is a paid service. It is currently free of charge until further notice.

Scenarios

You must add a domain name to DCDN before you can enable image editing. The image editing feature allows you to edit and cache images on DCDN nodes. This reduces the loads on origin servers.

Benefits

-
- **Reduced loads on origin servers**

  If images are edited and stored on origin servers, the images consume a lot of storage and compute resources on the origin servers. These increase the maintenance costs of your origin servers. The image editing feature allows you to edit and cache images on DCDN nodes. This reduces the loads on origin servers.
-
-

Configure image editing

Usage notes

DCDN allows you to edit images on edge nodes. You can set parameters to specify how you want to edit images. You must pass one or more actions such as `crop` or `rotate` to the `image_process` object to specify how you want to edit images. Separate multiple actions with forward slashes (/). DCDN edits images in order of actions. For example, `image_process=resize,w_200/rotate,90` specifies that the image is resized to a width of 200 pixels first, and rotates 90°.

Add parameters to the URL of an image

You can add image editing parameters to the end of the URL of an image.

- Format: `http://example.com/example.jpg?image_process=action,param_value`
  - `example.com`: the domain name accelerated by DCDN.
  - `example.jpg`: the name of the image.
  - `image_process`: the object to which you can pass image editing parameters.
  - `action,param_value`: the action, parameter, and value that specify how the image is to be edited. For more information, see Actions.
- Example: `http://example.com/example.jpg?image_process=resize,w_200/rotate,90`

Actions

You can specify one or more actions to perform on an image. The following table describes the actions that you can perform on images cached on DCDN nodes.

Feature	Action	Description
Convert image formats	format	Converts images to different formats.
Change image quality	quality	Adjusts the quality of images.
Crop images	crop	Crops images.
Resize images	resize	Resizes images.
Rotate images	• auto-orient: automatically rotates an image. • rotate: manually rotates an image.	auto-orient rotates images based on the orientation property. rotate rotates images clockwise based on the angle that you specify.
Change the color of an image	• bright: specifies the brightness of images. • contrast: specifies the contrast of images. • sharpen: specifies the sharpness of images.	Adjusts the brightness, contrast, and sharpness of images.
Manage image watermarks	watermark	Adds picture or text watermarks to images.
Query image information	info	Queries image information, including the width, height, format, and quality.

11.5.2. Convert image formats

Dynamic Route for CDN (DCDN) can automatically convert image formats to WebP or covert images to a specified format. You can convert images that are stored on DCDN nodes to a specified format. This topic describes how to convert image formats and provides examples.

> Note
> - Before you use image editing, you must to activate this feature.
> - Image editing is a paid service. It is currently free of charge until further notice.

Automatically convert images to WebP

DCDN determines whether to convert images to the WebP format based on the Accept header in a request. If `image/webp` is included in the Accept header, DCDN automatically converts images to the WebP format.

11.5.3. Change image quality

You can adjust the quality of an image by compressing the image, changing the absolute quality value, or changing the quality coefficient. Dynamic Route for CDN (DCDN) allows you to compress images stored on DCDN nodes without changing the image format.

> Note
> - Before you use image editing, you must to activate this feature.
> - Image editing is a paid service. It is currently free of charge until further notice.

Compress images

You can compress an image without resizing the image or changing the image format. Image compression compromises image quality but helps reduce data transfer costs.

11.5.4. Crop images

Dynamic Route for CDN (DCDN) allows you to crop images based on a specified size. This topic describes how to crop images and provides examples.

> Note
> - Before you use image editing, you must to activate this feature.
> - Image editing is a paid service. It is currently free of charge until further notice.

11.5.5. Resize images

Dynamic Route for CDN (DCDN) allows you to resize images. This topic describes how to resize images and provides examples.

> **Note**
> - Before you use image editing, you must to activate this feature.
> - Image editing is a paid service. It is currently free of charge until further notice.

11.5.6. Rotate images

Dynamic Route for CDN (DCDN) supports automatic and manual image rotation. You can set DCDN to automatically rotate images to a proper orientation or rotate images to a specified orientation. This topic describes how to rotate images and provides examples.

> **Note**
> - Before you use image editing, you must to activate this feature.
> - Image editing is a paid service. It is currently free of charge until further notice.

Automatic rotation

Images taken by some cameras carry the orientation property. Automatic rotation is applicable to only images that carry the orientation property. After you enable the automatic rotation feature, DCDN automatically rotates these images to a proper orientation.

> **Note**

Set the action to `auto-orient`.

Examples

```
image_process=auto-orient
```

11.5.7. Change the color of an image

The color of an image includes the brightness, contrast, and sharpness. Dynamic Route for CDN (DCDN) allows you to change the brightness, contrast, and sharpness of images stored on DCDN nodes. This topic describes how to change the color of images and provides examples.

> **Note**
> - Before you use image editing, you must to activate this feature.
> - Image editing is a paid service. It is currently free of charge until further notice.

11.5.8. Manage image watermarks

Dynamic Route for CDN (DCDN) supports picture watermarks and text watermarks. You can add picture watermarks and text watermarks to images. This topic describes how to add watermarks to images and provides examples.

> **Note**
> - Before you use image editing, you must to activate this feature.
> - Image editing is a paid service. It is currently free of charge until further notice.

Picture watermarks

Feature	Description	Parameter	Valid value
Watermark URL	You can specify watermark URLs that are accessible from the Internet. If authentication or permissions are required to access the specified URL, DCDN may fail to retrieve the watermark. Watermark URLs must be encoded in Base64. For more information, see Watermark encoding.	image	A Base64-encoded string.

Text watermarks

Feature	Description	Parameter	Valid value
Text content	Specifies the content of a text watermark. The text content must be encoded in Base64. For more information, see Watermark encoding.	text	A Base64-encoded string that contains at most 60 characters in length.
Text font	Specifies the font of a text watermark. The font name must be encoded in Base64. For more information, see Watermark encoding.	type	Up to 10 fonts are supported. For more information, see Text fonts. **Note** If you use a font that is not included in the 10 fonts, the font is recognized as the default font alihyaihei.
Text color	Specifies the color of a text watermark.	color	RGB color codes. For example, 000000 represents black and FFFFFF represents white. Default value: 000000.
Text rotation	Specifies the angle to which the text is rotated clockwise.	rotate	Supported angles are 90°, 180°, and 270°.

Feature	Description	Parameter	Valid value
Text tiling	Specifies whether to tile an image with text watermarks.	fill	Valid values: 0 and 1. Default value: 0. • 0: does not tile an image with text watermarks. • 1: tiles an image with text watermarks.

The following table describes the text fonts that are supported by text watermarks.

Text fonts

Text font	Description	Code
alihyaihei	A bold font. This is the default font.	YWxpaHlhaWhlaQ
hysong	A Songti font variant.	aHlzb25n
hyhei	A Heiti font variant.	aHloZWk
hyshuangxian	A double line font.	aHlzaHVhbmd4aWFu
fzltzhk	A Heiti font variant.	ZnpsdHpoaw
fzshengsks	A regular script font.	ZnpzaGVuZ3Nrcw
fzqusongjian	A Songti font variant.	ZnpxdXNvbmdqaWFu
zzgfxingyan	An artistic font.	enpnZnhpbmd5YW4
comfortaa	Comfortaa	Y29tZm9ydGFh
notosans	NotoSans	bm90b3NhbnM

Watermark positions

Watermark encoding

11.5.9. Query image information

This topic describes how to query image information and provides examples.

Parameters

Set the action to `info`.

The image information is returned in JSON format. The information includes the height, width, format, quality, and orientation of the specified image.

```
{
  "Length":1055089,
  "Width":1920,
  "Height":1080,
  "Quality":100,
  "Format":"JPEG",
  "Orientation":"UNDEFINED"}
```

Examples

```
example.com/image01.png?image_process=info
```

11.6. Configure the parameter filtering feature

The parameter filtering feature enables Dynamic Route for CDN (DCDN) to delete the query string in the URL of a request after DCDN receives the request. A query string contains parameters that follow the question mark `(?)` in a request URL.This feature increases the cache hit ratio and reduces the number of times that Dynamic Route for CDN (DCDN) retrieves resources from the origin. Therefore, this feature reduces data transfer costs and improves the efficiency of content delivery. This topic describes how to configure the parameter filtering feature.

Context

- Enable the parameter filtering feature

  A request URL may carry a query string that follows a question mark `(?)` ,for example, `http://alibaba.com/content?a` . If the query string is not essential for retrieving the requested content, we recommend that you enable the parameter filtering feature. After you enable this feature, DCDN automatically deletes the query string that follows the question mark `(?)` in request URLs. This increases the cache hit ratio.

  For example, the first time you retrieve `http://www.****.com/image` , the resource is fetched from the origin server instead of a DCDN node. Next time you retrieve `http://www.****.com/image?test1` , DCDN deletes the query string that follows the question mark `(?)` because the parameter filtering feature is enabled. The request directly hits the cached resource `http://www.****.com/image` .

- Disable the parameter filtering feature

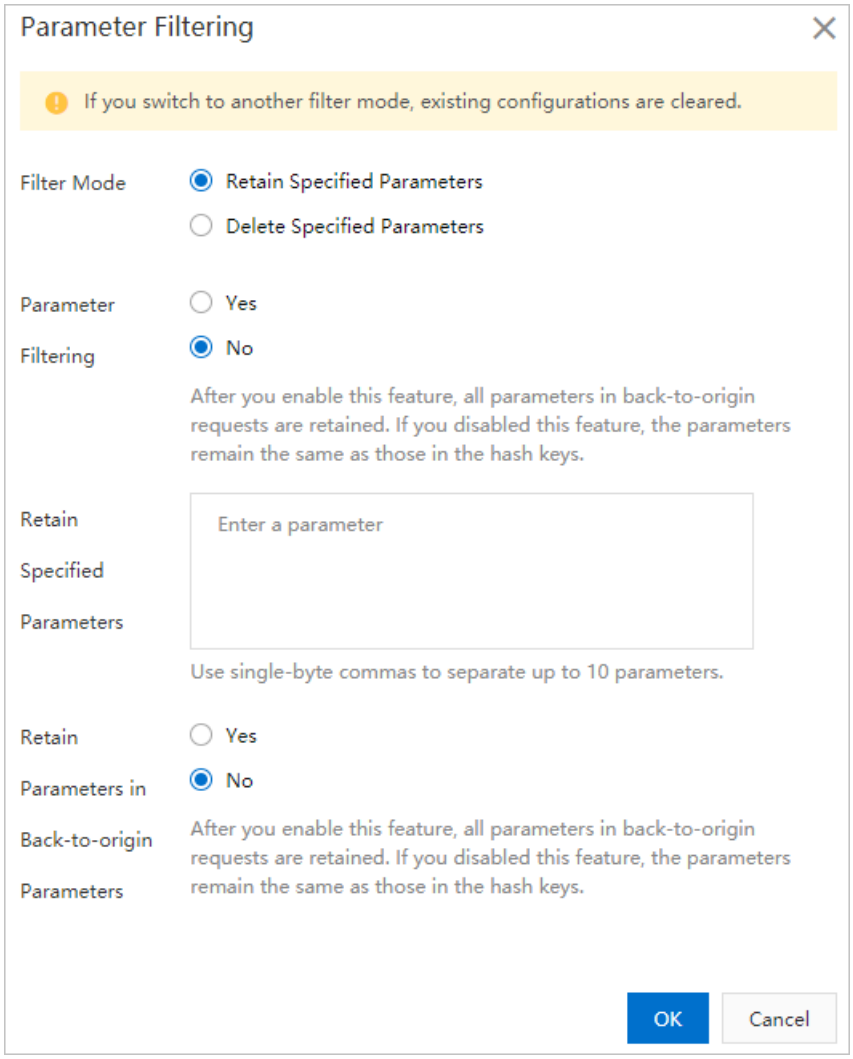
  A request URL may carry a query string that follows a question mark `(?)` . If the query string specifies important content, we recommend that you disable the parameter filtering feature. After the parameter filtering feature is disabled, the query string that follows the question mark `(?)` in a request URL must exactly match that of the cached version. Exact matches increase the accuracy of content retrieval.

For example, the first time you retrieve `http://www.****.com/image`, the resource is fetched from the origin server instead of a DCDN node. Next time you retrieve `http://www.****.com/image?test1`, the query string that follows the question mark `(?)` in the URL must exactly match that of the cached resource.Otherwise, DCDN does not return the cached version `http://www.****.com/image`. Instead, DCDN retrieves the requested resource `http://www.****.com/image?test1` from the origin server.

> **Note** The URL authentication feature has a higher priority over the parameter filtering feature. The authentication information in type A contains the parameters of an HTTP request. Therefore, a DCDN node must verify the signed URL of the request before the DCDN node caches a version of the requested resource. For more information about how to configure URL authentication, see Configure URL authentication.

Procedure

1.
2.
3.
4.
5. In the **Modify** section, Click **Parameter Filtering**. The following figure shows how to configure parameter filtering.



Note If you change the filtering mode, the current configuration is cleared.

You can select **Retain Specified Parameters** or **Delete Specified Parameters** for **Filter Mode**. The following table describes the parameters.

Filtering mode	Parameter	Description
	Parameter Filtering	◦ **Yes**: DCDN deletes the query string that follows the question mark `(?)` in request URLs. This increases the cache hit ratio. **Note** If you enable **Parameter Filtering** and do not specify parameters in **Retain Specified Parameters**, all parameters that follow the question mark `(?)` are deleted. ◦ **No**: The query string that follows the question mark `(?)` in a request URL must exactly match that of a cached version. Exact matches increase the accuracy of content retrieval.

Filtering mode	Parameter	Description
Retain Specified Parameters	**Retain Specified Parameters**	Specify the parameters that you want to retain. You can specify up to 10 parameters. Separate multiple parameters with commas (,). **Note** If you specify only **Retain Specified Parameters**, the parameter filtering setting does not take effect. This parameter must be specified together with **Parameter Filtering** and **Retain Parameters in Back-to-origin Parameters**. Examples: ◦ Example 1: Only **Parameter Filtering** is enabled. **Retain Parameters in Back-to-origin Parameters** is disabled. Original URL: http://example.com/image_01.png?key1=123&key2=321 Cached key: http://example.com/image_01.png Back-to-origin URL: http://example.com/image_01.png ◦ Example 2: **Parameter Filtering** is enabled, and key1 is specified in **Retain Specified Parameters**. Original URL: http://example.com/image_01.png?key1=123&key2=321 Cached key: http://example.com/image_01.png?key1=123 Back-to-origin URL: http://example.com/image_01.png?key1=123 ◦ Example 3: **Parameter Filtering** and **Retain Parameters in Back-to-origin Parameters** are enabled. Original URL: http://example.com/image_01.png?key1=123&key2=321 Cached key: http://example.com/image_01.png Back-to-origin URL: http://example.com/image_01.png?key1=123&key2=321 ◦ Example 4: **Parameter Filtering** and **Retain Parameters in Back-to-origin Parameters** are enabled. Key1 is specified in **Retain Specified Parameters**. Original URL: http://example.com/image_01.png?key1=123&key2=321 Cached key: http://example.com/image_01.png?key1=123 Back-to-origin URL: http://example.com/image_01.png?key1=123&key2=321

Filtering mode	Parameter	Description
	Retain Parameters in Back-to-origin Parameters	○ **Yes**: DCDN retains the entire query string in a request URL when it forwards the request back to the origin server. ○ **No**: DCDN retains only the specified parameters in a request URL when it forwards the request back to the origin server.
Delete Specified Parameters	**Delete Specified Parameters**	Specifies the parameters that you want to delete. You can specify up to 10 parameters. Separate multiple parameters with space characters. Example: Key1 is specified in **Delete Specified Parameters** and **Retain Parameters in Back-to-origin Parameters** is enabled. Original URL: http://example.com/image_01.png?key1=123&key2=321 Cached key: http://example.com/image_01.png?key2=321 Back-to-origin URL: http://example.com/image_01.png?key1=123&key2=321
	Retain Parameters in Back-to-origin Parameters	○ **Yes**: DCDN retains the entire query string in a request URL when it forwards the request back to the origin server. ○ **No**: DCDN deletes the specified parameters in a request URL when it forwards the request back to the origin server.

6. Click **OK**.

11.7. Configure video seeking

Video seeking allows you to seek a specified position without compromising the playback quality when you play video and audio content. This topic describes how to configure video seeking.

Context

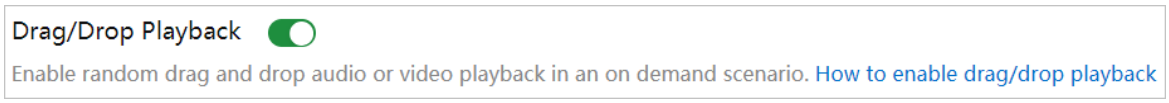
After video seeking is enabled, if a client seeks a specified position when it plays video or audio on demand, the client sends a request to the server. The request contains the URL of the video or audio file, such as `http://www.aliyun.com/test.flv?start=10`. The start parameter specifies the position that you want to seek. In the example, the specified position is the tenth byte. After the server receives the request, the server seeks the keyframe at the specified position and then returns the content that starts from this keyframe. If no keyframe can be found at the specified position, the server seeks the last keyframe before the specified position.

- Before you configure video seeking, make sure that the origin server supports HTTP range requests. If an HTTP request header contains the Range field, the origin server must return the following status message: 206 partial content.
- The following table describes the sample URLs and the file formats that are supported by video seeking.

Format	Meta information	start parameter	Example
MP4	The meta information about a video file on the origin server must be contained in the file header and cannot be contained in the file tail.	The start parameter specifies the start time. The start time is measured in seconds. Decimals are supported to indicate milliseconds. For example, start=1.01 indicates that the video is played from 1.01 seconds. If the frame at the position that is specified by the start parameter is not a keyframe, Dynamic Route for CDN (DCDN) locates the last keyframe before that position.	The request URL `http://domain/video.mp4?start=10` indicates that the video is played from the tenth second.
FLV	Video files on origin servers must contain meta information.	The start parameter specifies the byte. If the byte that is specified by the start parameter is not a keyframe, DCDN automatically locates the last keyframe before that byte.	The request URL `http://domain/video.flv?start=10` indicates that a video is played from the tenth byte. If the tenth byte is not a keyframe, the video is played from the last keyframe before the tenth byte.

Procedure

1.
2.
3.
4.
5. In the **Drag/Drop Playback** section, turn on **Drag/Drop Playback**.



12.Security Settings

12.1. Configure bot traffic management

To prevent your business from malicious traffic or crawlers, Alibaba Cloud Dynamic Route for CDN (DCDN) provides the bot traffic management feature. This feature is integrated with AI intelligent protection and uses information such as authorized crawlers and threat intelligence to identify and block malicious requests. This topic describes how to enable and configure bot traffic management.

Context

Bot traffic management is enabledin the DCDN console.To enable bot traffic management, .

Procedure

1.
2.
3.
4. In the left-side navigation pane on the details page of the specified domain name, click **Security Settings**.
5. On the **Bot Traffic Management** tab, set **Authorized Crawlers**, **Threat Intelligence**, and **AI Intelligent Protection**.

Parameter	Description
Authorized Crawlers	You can enable or disable this feature. **Note** Authorized crawlers function as a whitelist that contains authorized search engines. Requests from authorized crawlers are allowed to access the domain name. You can click **Modify** to view, enable, or disable authorized crawlers.
Threat Intelligence	You can enable or disable this feature. **Note** Threat intelligence provides information about bot traffic, such as fingerprints, IP libraries, and proxy IP addresses. You can enable threat intelligence after simple configurations to block malicious requests. You can click **Modify** to view, enable, or disable threat intelligence.

Parameter	Description
AI Intelligent Protection	You can enable or disable this feature. **Note** AI intelligent protection automatically analyzes and studies workloads based on algorithms, generates bot traffic fingerprints, and creates protection rules against malicious requests. You can click **Modify** to view, enable, or disable AI intelligent protection.

12.2. Configure precise access control

This topic describes the precise access control feature, and how to enable and configure this feature.

Overview

The precise access control feature allows you to add custom match conditions to match user requests and perform specified actions on requests that matches the conditions. Match conditions support common HTTP fields such as IP, URL, and header. You can add different match conditions to meet the protection requirements in different scenarios.

ACL rules

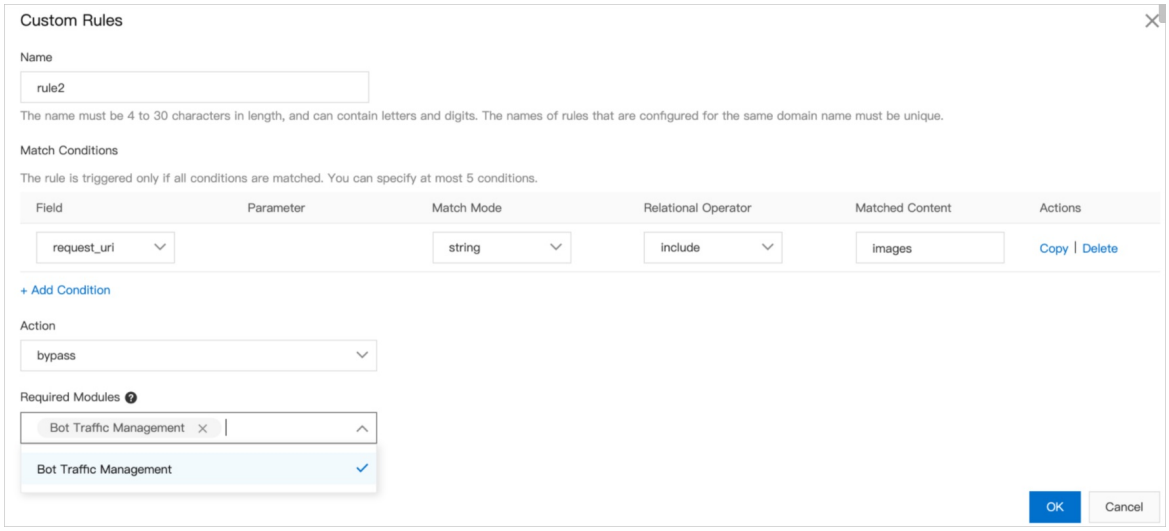
An access control list (ACL) rule consists of one or more match conditions and one action. You can add one or more ACL rules. If you add multiple ACL rules, the rules are listed and matched against requests in descending order of priority. When a rule is matched, the system stops matching subsequent rules.

Enable precise access control

To enable the precise access control feature, visit the Contact Sales page, and leave your contact information. An Alibaba Cloud sales representative will contact you as soon as possible.

Procedure

1.
2.
3.
4. Click **Security Settings** and select the **Precise Access Control** tab.
5. Add an ACL rule.



Match conditionsSpecifies the HTTP field of the request to match.

> **Note**
>
> - You can add one or more match conditions. If you add multiple match conditions, the ACL rule is triggered only if all conditions are matched.
> - A match condition consists of Field, Parameter, Match Mode, Relational Operator, and Match Content. When you configure a match condition, parameters that cannot be configured are not used in the match condition. You can ignore these parameters.

The following table describes the parameters of a match condition, such as Field and Relational Operator.

Field	Parameter	Match mode	Relational operator	Matched content
requst_uri	N/A	RegEx	match or NotMatch	String
		string	include, exclude, equal, or NotEqual	String
header	Request header	RegEx	match or NotMatch	String
		string	include, exclude, equal, or NotEqual	String
		string	NotExist	N/A
method	N/A	string	equal or NotEqual	String

Field	Parameter	Match mode	Relational operator	Matched content
ip	N/A	string	in or NotIn	IP addresses that are separated with commas (,).
referer	N/A	RegEx	match or NotMatch	String
		string	include, exclude, equal, or NotEqual	String
user-agent	N/A	RegEx	match or NotMatch	String
		string	include, exclude, equal, or NotEqual	String
cookie	N/A	RegEx	match or NotMatch	String
		string	include, exclude, equal, or NotEqual	String
content-type	N/A	RegEx	match or NotMatch	String
		string	include, exclude, equal, or NotEqual	String
x-forwarded-for	N/A	RegEx	match or NotMatch	String
		string	include, exclude, equal, or NotEqual	String
post-body	N/A	RegEx	match or NotMatch	String
		string	include, exclude, equal, or NotEqual	String
params	N/A	RegEx	match or NotMatch	String
		string	include, exclude, equal, or NotEqual	String

ActionThe action that is performed when a request matches the conditions that you configure. Valid values:

- *observe*: Requests that match the configured conditions are allowed and recorded in the log. These requests carry a header when they are redirected to the origin server. This header defines the risk level of these requests and helps the origin server process these requests.
- *block*: Requests that match the configured conditions are rejected. A 403 status code is returned.
- *bypass*: Requests that match the configured conditions are allowed. You need to select a required module. The selected module will process requests that match the configured conditions. Modules that are not selected allow these requests.

6. After the ACL rule is configured, click **OK**.
7. (Optional) You can add multiple ACL rules and adjust the priority of the rules that you add.

13.Advanced Settings

13.1. Configure IPv6

This topic describes how to enable the IPv6 feature in the Alibaba Cloud Dynamic Route for CDN (DCDN) console. After you enable IPv6 in the console, IPv6 clients can send IPv6 requests to DCDN. DCDN can include the IPv6 information of the clients in back-to-origin requests.

Context

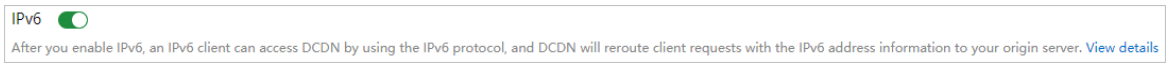
Most of the DCDN nodes support IPv6. You can enable this feature for an accelerated domain when you configure features for the domain.

After IPv6 is enabled, clients can send IPv6 requests to the DCDN nodes that are nearest to the clients over an IPv6 network if these DCDN nodes support IPv6. If the nearest DCDN nodes do not support IPv6, the clients can access the DCDN nodes by sending IPv4 requests.

> **Note** The DCDN nodes in China (Hong Kong), China (Macao), China (Taiwan), and regions outside China do not support IPv6.

Procedure

1.
2.
3.
4. In the left-side navigation pane of the page that appears, click **Advanced Settings**.
5. Turn on **IPv6**.

   After you enable IPv6, a client can send IPv6 requests to DCDN nodes. Requests that are redirected from DCDN nodes to your origin server can include the IPv6 information of the client.

14.WebSocket

14.1. What is WebSocket?

This topic describes the concept, benefits, and application scenarios of WebSocket.

Notice

Overview

WebSocket is a new network protocol that enables interaction between a web browser and a web server over a persistent Transmission Control Protocol (TCP) connection. WebSocket supports full-duplex communications that allow the server to actively send data to the client. Therefore, WebSocket requires only one handshake to establish a bi-directional, full-duplex, and persistent connection between the browser and the server. This simplifies the data exchanges between the client and the server.

Benefits

Many websites are using Asynchronous JavaScript and XML (AJAX) polling to implement push technologies. Based on the polling technique, the browser sends HTTP requests to the server at specific intervals, such as every second. Then, the server returns the most recent data to the browser of the client.

The disadvantage of this model is that the browser has to continuously send requests to the server. HTTP requests may have a large header and a small payload. The HTTP requests of this type result in a waste of bandwidth and other resources. The WebSocket protocol that is defined by HTML5 has the following benefits:

- Each message that is exchanged between the client and the server contains a small header. The size of the header is about 2 bytes.
- Instead of returning data after receiving a request from the browser, the server actively pushes data to the browser when new data is available.
- The WebSocket protocol helps you minimize the usage of server and bandwidth resources, and facilitate real-time communication.

Scenarios

Scenario	Description
Live commenting	User A sends a live comment through a mobile phone and wants to use the mobile phone to view the live comments that are sent by other clients. To meet the requirements, you can use WebSocket to push the live comments that are sent by other clients to the mobile phone of User A. Then, User A can view the live comments that are sent by other users.
Online education	When a teacher offers courses to students online, the teacher can use a client to send data to the clients of the students in real time based on WebSocket communication. The examples of the sent data include notes and outlines.

Scenario	Description
Real-time quotes for financial products	To handle the fluctuating prices of financial products, such as stock and gold, WebSocket pushes the up-to-date prices to the clients of global traders in real time. This helps the traders make informed decisions at the earliest opportunity.
Live sportscast	Live sportscasts are the top concern for a large number of sports lovers all over the world. WebSocket allows for real-time updates in live sportscasts to ensure optimal user experience.
Video conferencing	Video conferencing is widely used in diverse scenarios. In a video conference, WebSocket helps to deliver real-time information to participants who join the conference through multiple ends.
Location-aware applications	An increasing number of developers apply the GPS feature of mobile devices to location-aware applications. You can use WebSocket to keep tracking the location of a user. For example, your WebSocket-based application can record the movement trails of the user. This allows you to collect more details about the user.

14.2. Configure WebSocket

The WebSocket protocol simplifies data exchange between a client and a server by allowing the server to push data to the client. WebSocket increases the utilization of server and bandwidth resources and reduces latency. This topic describes how to enable and configure WebSocket.

> **Notice**
>
> - By default, WebSocket is disabled. After you enable the WebSocket feature, the feature is displayed in the Dynamic Route for CDN (DCDN) console.
>
>   If you want to enable WebSocket, submit a ticket.
> - WebSocket is used only for dynamic content delivery.

The following sections describe how to enable and use the WebSocket feature:

- Apply to use WebSocket
- Enable WebSocket
- Query bandwidth usage and HTTP status codes
- Disable WebSocket

Apply to use WebSocket

Apply to use WebSocket: You can submit a ticket and apply to use WebSocket. Only enterprise users can apply to use WebSocket. You cannot enable WebSocket for domain names whose origin servers are deployed outside the Chinese mainland and whose acceleration region is set to Mainland China Only or Global.

Alibaba Cloud customer service reviews the application and notifies you of the result by text messages and emails within one business day. After your application is approved, you can enable WebSocket in the console.

Notice WebSocket is a value-added feature and is billed as an independent service. For more information about WebSocket billing, see DCDN Pricing.

Enable WebSocket

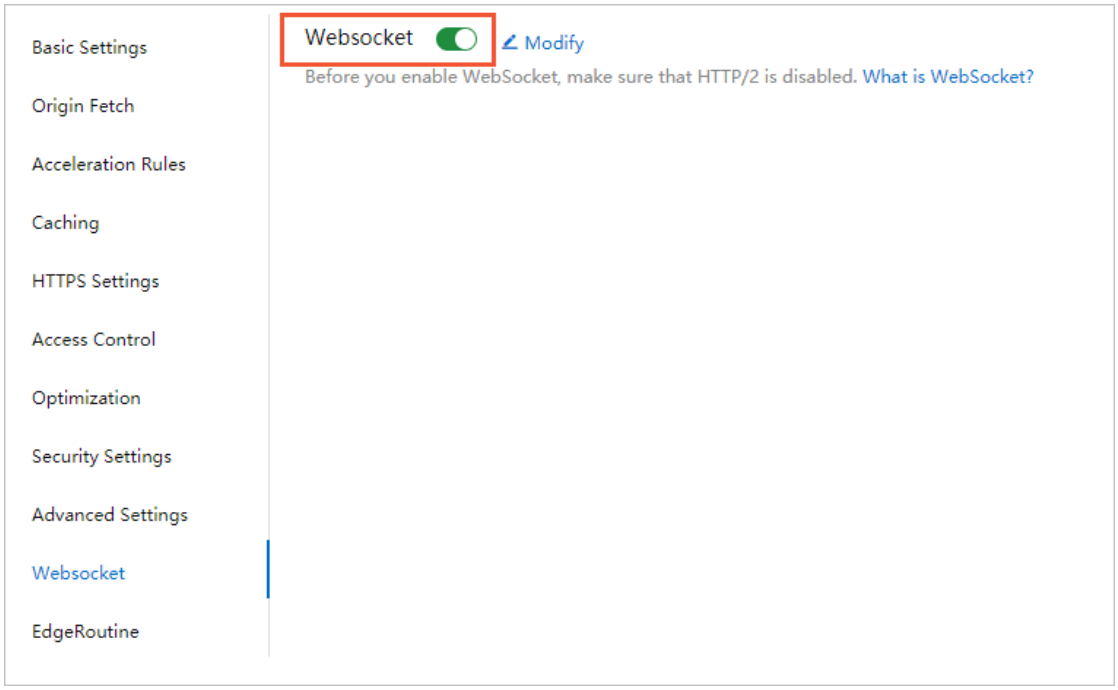
Note

Before you enable the WebSocket feature, make sure that the following requirements are met:

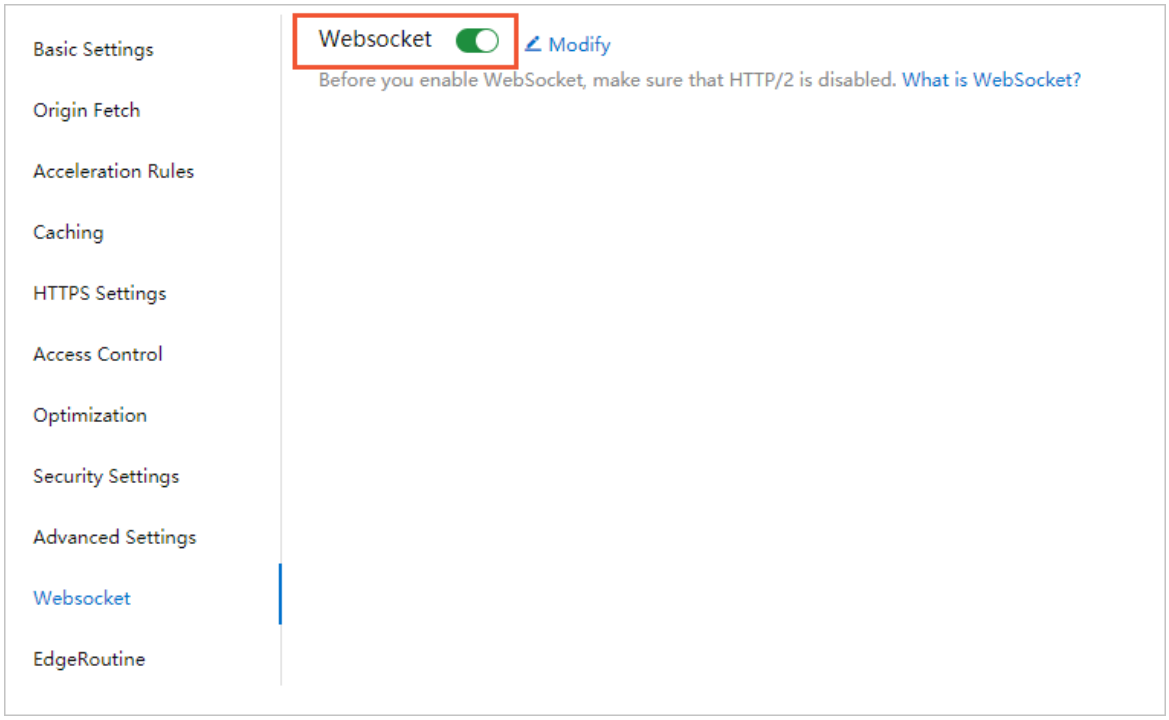
- Your Alibaba Cloud account has passed real-name verification for enterprises, and a domain name that belongs to your Alibaba Cloud account has a valid Internet Content Provider (ICP) number.
- HTTP/2 is disabled.

  If clients do not support HTTP/2, you must disable HTTP/2 for DCDN. Otherwise, service errors may occur. For more information, see Enable HTTP/2.

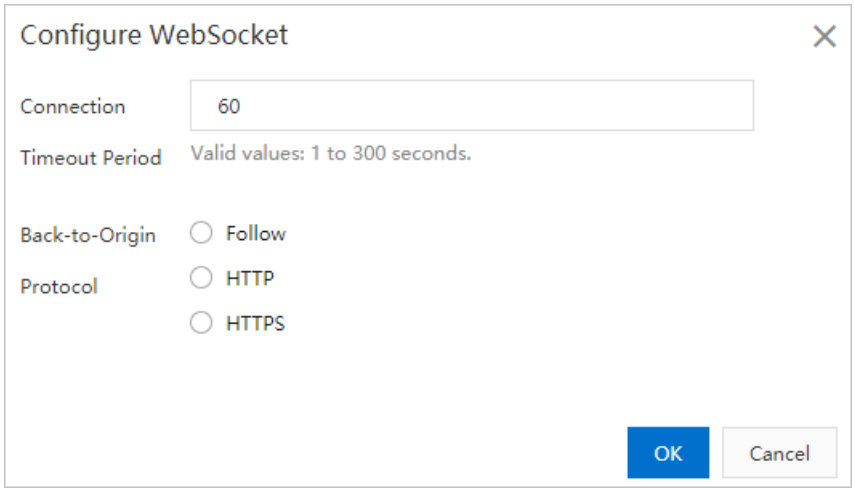
1.
2.
3.
4. (Optional)Turn on **Dynamic Acceleration**. If the feature is enabled, skip this step.
   i. In the left-side navigation pane of the domain name, click **Acceleration Rules**.
   ii. Turn on **Dynamic Acceleration**.



5. In the left-side navigation pane of the domain name, click **Websocket** and turn on **Websocket**.



6. Click **Modify**.
7. In the **Configure WebSocket** dialog box, configure the Connection Timeout Period and Back-to-Origin Protocol parameters.



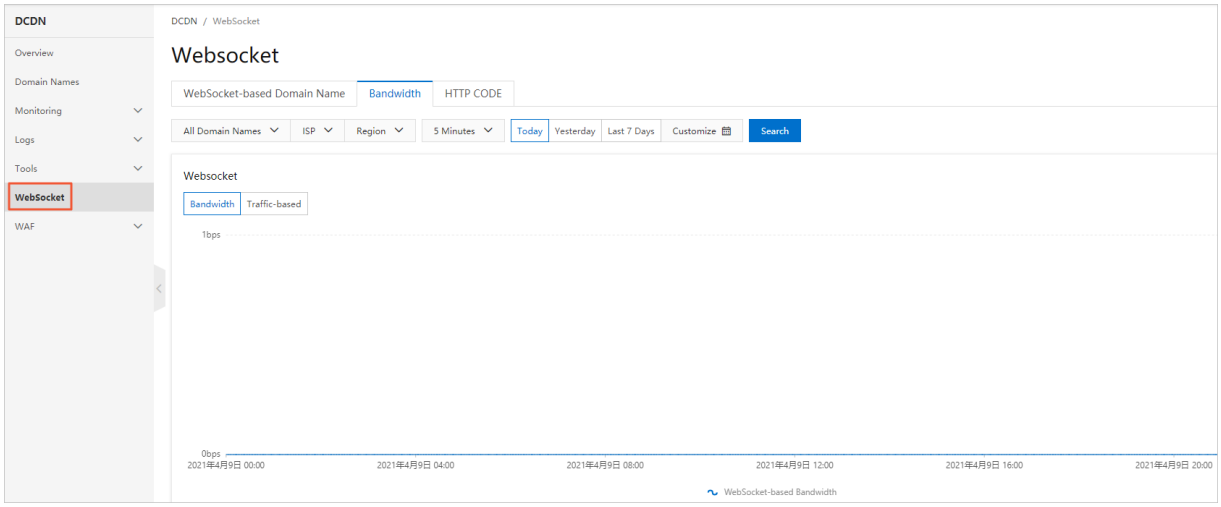
Parameter	Description

Parameter	Description
Connection Timeout Period	The timeout period is the interval at which the client sends data packets to the server to synchronize the current status. ○ Default value: 60. ○ Unit: seconds. ○ We recommend that you set the timeout period based on the following rule: A ≤ B ≤ C. ■ A: the timeout period of the clients. ■ B: the timeout period of DCDN. ■ C: the timeout period of the origin server. **Note** If the timeout period of the client is longer than that of the origin server, errors may occur.
Back-to-Origin Protocol	You can specify a protocol over which requests are redirected to the origin server. ○ **Follow**: DCDN uses the same protocol (HTTP or HTTPS) as the client to redirect requests to the origin server. **Port 443 or 80 of the origin server must be open.** ○ **HTTP**: DCDN redirects requests to the origin server over HTTP. ○ **HTTPS**: DCDN redirects requests to the origin server over HTTPS. **Port 443 of the origin server must be open.**

8. Click **OK**.

Query bandwidth usage and HTTP status codes

After you configure and use WebSocket, click **WebSocket** in the left-side navigation pane in the DCDN console to go to the Websocket page. Then, you can view monitoring information about **Bandwidth** and **HTTP CODE**.



Disable WebSocket

If you no longer want to use WebSocket, you can disable WebSocket in the DCDN console. The WebSocket is disabled immediately after you turn off the switch.