Edgecast

HTTP Large Administration Guide

edgecast

Disclaimer

Care was taken in the creation of this guide. However, Edgecast cannot accept any responsibility for errors or omissions. There are no warranties, expressed or implied, including the warranty of merchantability or fitness for a particular purpose, accompanying this product.

Trademark Information

EDGECAST is a registered trademark of Edgecast Inc.

WINDOWS and SILVERLIGHT are registered trademarks of Microsoft Corporation.

About This Guide

HTTP Large Administration Guide
Version 5.41
6/2/2022

© 2022 Edgecast Inc. All rights reserved.

Table of Contents

Data Delivery Platforms 1
Introduction 1
Choosing an HTTP Platform 2
HTTP Large Platform 3
How Does It Work? 3
Indicating Where Your Content Can Be Found 5
Accessing Your Content 6
CDN URL 7
Edge CNAME URL 9
How Requests are Handled 10
Phase 1: Request 10
Phase 2: Security 10
Phase 3: Check for Cached Content 12
Phase 4: Retrieval, Delivery, and Caching 14
Common HTTP Status Codes 15
Configuring an Origin Server 16
Overview 16
Customer Origin Server 17
Origin Configuration 18
Directory Name 18
Domain – IP Address Configuration 19
HTTP Host Header 21
Load Balancing 21
Origin Shield 22
Customer Origin Management 24
Response Headers 27

Firewall Access 27
Best Practices (Dynamic Application) 27
CDN Origin Servers 28
Masking the URL of a CDN Origin Server (CNAME) 29
Setting up DNS for an Edge CNAME Deactivation 31
Edge CNAME Deactivation 31
Securing Your Content 32
Overview 32
Secure HTTP Requests (HTTPS) 32
Certificate Provisioning System 32
Legacy HTTPS 39
Token-Based Authentication 41
Country Filtering 41
Administering Country Filtering Configurations 42
Protecting Assets Using Both Country Filtering and Token-Based Authentication 45
Cache Management 46
Default Cache Management 46
Caching & Incomplete Downloads (Complete Cache Fill) 47
Loading Assets 47
Automatically Loading Assets through the Web Services REST API 49
Purging Assets 50
Automated Purging Through the Web Services REST API 50
Manually Purging Assets 51
Purge and Load History 56
Query String Caching 57
Query String Logging 58
Asset Compression 58
Customer Origin Server Compression 58
Edge Server Compression 62
Overwriting Cache Headers 64
Bandwidth Throttling 65
Overview 65

Configuring Bandwidth Throttling 65
HTTP Streaming Solutions 67
Overview 67
Glossary 69

Data Delivery Platforms

Introduction

A platform is a term that encompasses the infrastructure of dedicated servers and devices that have been distributed across our worldwide network of points-of-presence (POPs) for the purpose of efficiently securing and delivering a particular type of data from an origin point to your customers. Our CDN provides several specialized platforms for data delivery. These platforms are listed below.

Platform	Protocol	Description
Application Delivery Network (ADN)	HTTP HTTPS	This platform has been optimized to quickly deliver dynamic content generated from your web applications.
HTTP Large	HTTP HTTPS	This platform has been optimized to cache large assets, which allows it to quickly deliver them to your customers.
HTTP Small	HTTP HTTPS	This platform has been optimized to index small and more frequently used assets for quicker retrieval and delivery.

Choosing an HTTP Platform

There are two platforms that specialize in serving static content over the HTTP and HTTPS protocols, which are HTTP Large and HTTP Small. The use of these platforms for serving data from your website is not mutually exclusive. In fact, an optimal configuration usually involves both of these platforms to serve your entire website's content. We will now take a closer look at both platforms.

Although the HTTP Large platform can serve any type of asset over the HTTP/HTTPS protocol, it specializes in the delivery of large assets, as well as those that are not downloaded as frequently. The size of an asset delivered by this platform is usually greater than 300 KB. The typical applications for this platform includes the delivery of videos, high resolution images, audio content, online games, software downloads, and software updates. This platform supports a variety of data delivery methods, such as by file, by chunk, or even byte serving.

Note: The HTTP Large platform supports video streaming using HTTP Progressive Download and Dynamic Cloud Packaging. For more information on these streaming solutions, please refer to the CDN Help Center.

Unlike the HTTP Large platform, the HTTP Small platform serves cached content straight from RAM. This reduces the amount of time spent on disk transactions whenever a customer views your web pages. Since assets are being served from RAM, it is highly recommended that you only serve assets that are requested frequently and that are smaller than 300 KB on this platform. This is why this platform is ideal for most website content, including websites that take advantage of ad serving, e-commerce, and photo sharing capabilities. The types of assets that most of these sites employ are small in nature and are requested frequently. For example, a typical website will require that a user download HTML, CSS, JS, and thumbnail image assets.

Important: It is crucial to use the correct platform for the type of content that you would like to serve. Choosing the wrong platform can produce less than optimal performance for your website. For detailed recommendations on which platform you should use to serve your assets, please contact your CDN account manager.

HTTP Large Platform

How Does It Work?

Serving content through the HTTP Large platform allows our CDN network to transmit your content in a more efficient manner to your customers. Our network has been optimized to allow data to quickly travel between where it is stored and the point-of-presence (POP) closest to the customer that requested it. Additionally, a copy of the requested content may be stored on that POP. This process is known as caching. Once content has been cached, all future requests for that content from that region can be delivered directly from that POP. This eliminates the need to retrieve the data from the origin server. The following diagram demonstrates this process.

Note: For the purpose of simplifying this example, only the North American portion of our worldwide network is displayed.

CDN Data Delivery

The above diagram demonstrates how content is delivered via the CDN to a user. The steps involved in this process are listed below.

Note: Only content stored in our CDN storage solution or on a customer origin server may be served through our network.

1. This data delivery process starts with a user's request for content. This request is directed to the POP closest to the user.
2. An edge server in that POP will check whether the requested content has been previously cached.
    - **Cached:** The asset is immediately served up to that client. In the above diagram, the dashed white lines indicate the clients that are able to take advantage of a previously cached asset and thus are able to enjoy even faster downloads.
    - **Not Cached:** The POP will forward the request to the appropriate CDN or customer origin server. The origin server will respond with the requested content.
3. The requested asset is delivered via the POP from step 1 to the client that requested it.
    - In the above diagram, there are four clients requesting content. These clients are located close to Seattle and Atlanta. The Seattle and Atlanta clients will receive the asset through the Seattle POP and Atlanta POP, respectively. This is represented by orange lines in the diagram. Sending an asset through our network and bypassing traditional Internet communication routes ensures the efficient transmission of your asset and reduced bandwidth load on your network.
    - The last leg involved in the delivery of the requested content requires the client's ISP to deliver the asset from our POP. This is indicated by white lines in the above diagram.
4. The requested asset may then be cached on the POP that received and fulfilled the request. This allows that POP to immediately fulfill future requests for that asset without having to forward the request to the origin server.

Indicating Where Your Content Can Be Found

Before assets can be served through our CDN to your customers, they must first be retrieved from a web server. You can either store your assets on your own web server (i.e., customer origin server) or on a storage server (i.e., CDN origin server) in our CDN. Below you will find a brief discussion on the advantages for each type of origin server.

Customer Origin Server

The main advantage of a customer origin server is that you do not have to transfer data to another server. All of your data can remain as it is currently structured. This can save you the amount of time it takes to upload and maintain that data on a separate server. Additionally, you can configure the cache instructions that will be assigned to your assets on your web server without having to purchase HTTP Rules Engine.

CDN Origin Server

The main advantage of a CDN origin server is that this type of server is inside our CDN network. Your data is no longer susceptible to network issues due to denial of service attacks or high traffic volume. This guarantees that the data will be available for distribution to your clients. Additionally, since the data does not have to be retrieved from a location outside our network, it can be delivered even faster to your clients.

Accessing Your Content

Regardless of whether you have stored your assets within our network, you will need to access your assets using a URL that points to our network. This type of URL allows your content to take advantage of our worldwide content delivery network. Your data will travel through our network until it reaches the point-of-presence (POP) closest to the client that requested the asset.

By default, your content will be cached there according to the instructions found in each asset's Cache-Control and Expires headers. These headers will also determine how the asset will be cached on a client's HTTP client (e.g., browser). If an asset does not have Cache-Control and Expires header information, then the default configuration will cause the asset to be cached for 7 days. Caching assets at our POP allows users served in that region to retrieve the asset without having to wait for it to be transferred from a CDN or customer origin server. This ensures much quicker data delivery. For more information on how assets are cached, please refer to the **Cache Management** chapter.

There are two types of URLs that can be used to request content through our CDN, which are:

- **CDN URL:** A system-defined URL that identifies a platform/service and a location/asset that will be routed through our network.
- **Edge CNAME URL:** A URL that uses an edge CNAME and a CNAME record to mask a CDN URL. This type of URL is more user-friendly than a CDN URL.

Both types of URLs can be used to identify an asset that will be routed through our content delivery network. The main difference between these two URLs is that an edge CNAME URL allows you to use a custom domain (e.g., www.mydomain.com) to reference content that will be routed through our network.

Important: CDN and edge CNAME URLs are case-sensitive. Using an improper case in the URL will result in a 404 – Not Found error message.

Tip: View a base URL for both CDN and edge CNAME URLs from the **HTTP Large Object** page.

In the following example, an edge CNAME called "www.mydomain.com" has been set to mask the following segment of a CDN URL "wpc.0001.edgecastcdn.net/800001/Business." Both of the following URLs request the same asset (i.e., Graphs.xml).

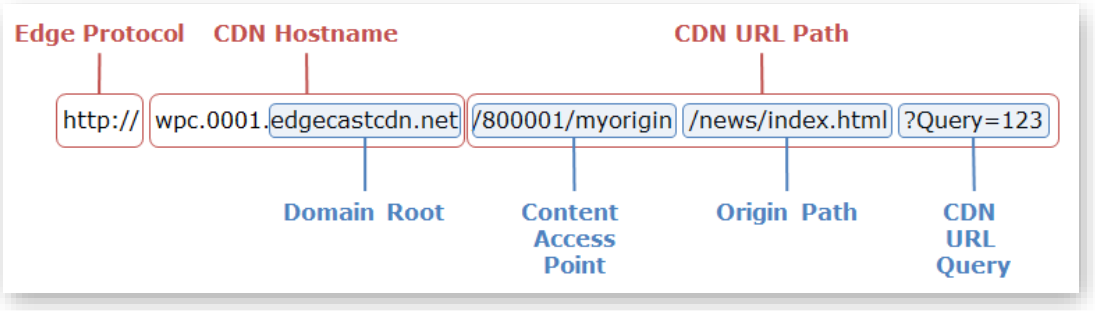
URL Type	URL
CDN URL	http://wpc.0001.edgecastcdn.net/800001/Business/Documents/Graphs.xml
Edge CNAME URL	http://www.mydomain.com/Documents/Graphs.xml

CDN URL

A CDN URL is any URL that points to our CDN and contains a content access point. Each customer account contains at least one CDN URL per platform. A CDN URL will be assigned to your account for the following items:

- CDN storage (root folder)
- Each customer origin configuration (root folder)
- Each edge CNAME configuration
- Each additional service that has been activated on this platform.

The following diagram indicates the different components in a CDN URL. Keep in mind that *xxxx* represents your CDN account number.

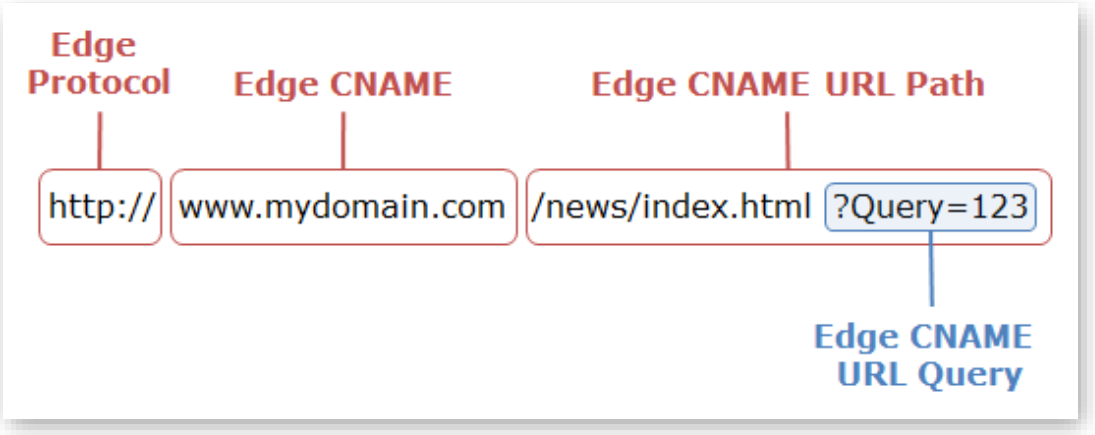


Name	Description
Edge Protocol	The edge protocol is always either "HTTP" or "HTTPS" for the HTTP Large platform.
CDN Domain	The CDN domain is the unique domain used by our CDN to determine the platform and the CDN customer associated with a request. In the above example, the CDN domain is "wpc.*xxxx*.edgecastcdn.net," where "wpc" identifies the HTTP Large platform and *xxxx* stands for your CDN account number.
CDN URL Path	The CDN URL path is the portion of the URL that appears after the CDN domain. The CDN URL query portion of the URL is optional.
Domain Root	The domain root identifies the base CDN domain that has been assigned to your CDN account. In the above example, the domain root is edgecastcdn.net.

Name	Description
Content Access Point	The content access point, which appears directly after the CDN domain, allows our CDN to identify the source for the requested content. A content access point can be one of the following for the HTTP Large platform: • **CDN origin server:** /00*xxxx* • **Customer origin server:** /80*xxxx*/*CustomerOrigin* • **Live streaming (Dynamic Cloud Packaging):** /20*xxxx* • **Live streaming playback (Dynamic Cloud Packaging):** /24*xxxx* • **CDN origin server playback (Dynamic Cloud Packaging):** /04*xxxx* • **Customer origin server playback (Dynamic Cloud Packaging):** /84*xxxx* The two digit number that precedes the account number (i.e., *xxxx*) is called the origin identifier. For the purposes of the HTTP Large platform, it instructs our CDN as to where traffic should be routed (as indicated above). The term *CustomerOrigin* represents the name assigned to the desired customer origin configuration.
Origin Path	The origin path indicates the location of the desired asset. This path starts from the root folder of the origin server identified by the content access point.
CDN URL Query	Optional. A query string can be appended at the end of the CDN URL. By default, query strings are ignored for the purposes of caching and event logging.

Edge CNAME URL

An edge CNAME URL is any URL that uses an edge CNAME to mask a CDN URL. The following diagram indicates the different components in an edge CNAME URL.



Name	Description
Edge Protocol	The edge protocol is always either "HTTP" or "HTTPS" for the HTTP Large platform.
Edge CNAME	There is a CDN and a DNS aspect to an edge CNAME. The CDN aspect is an edge CNAME configuration that allows our servers to identify a contact access point. In addition to this configuration, a DNS server must be configured to recognize the alias for the CDN URL. The alias set on the DNS server should match the domain (e.g., wpc.*xxxx*.edgecastcdn.net) associated with the edge CNAME configuration. If a directory path has been specified in the edge CNAME configuration, then our servers will rewrite the URL to point to the appropriate folder.
Edge CNAME URL Path	The edge CNAME URL path indicates the location of the desired asset on the origin server identified by the edge CNAME. The starting point for this path is determined by the relative path associated with the corresponding edge CNAME configuration. If a relative path has not been defined in the edge CNAME configuration, then the edge CNAME URL path starts from the root folder of the origin server.
Edge CNAME URL Query	Optional. A query string can be appended at the end of the edge CNAME URL. By default, query strings are ignored for the purposes of caching and event logging.

How Requests are Handled

The following steps take place when a user requests an asset:

1. Request
2. Security
3. Check for Cached Content
    - Check for Content Freshness
4. Retrieval, Delivery, and Caching

Below you will find an explanation for each phase in this procedure.

Phase 1: Request

Before an asset can be delivered to a client, a request must be made to our HTTP Large servers. Making a request to one of our HTTP Large servers is simply a matter of using a CDN or edge CNAME URL that points to the desired asset. The following sample CDN URL points to an asset that will be delivered through the HTTP Large platform:

http://wpc.0001.edgecastcdn.net/000001/Graphs.xml.

This request will be handled by the point-of-presence (POP) closest to that client. This is very important for the purposes of caching, since cached versions of an asset are specific to each POP. If the requested content has not been cached on that POP, then it will need to request it from an origin server or an origin shield server.

Phase 2: Security

The first check performed by an edge server in a POP is to find out whether the requested content has been secured. We offer the following methods for securing your content:

- Country Filtering
- Token-Based Authentication
- HTTP Rules Engine

If the request fails to meet any of the above security requirements, then our edge servers will deny the request. No additional steps will be taken.

Country Filtering

Country Filtering allows you to secure content by directory. If the requested content resides in a secured directory, then the location of the user requesting the content determines whether the request will be authorized. If the request is denied, then the HTTP client will receive a 403 Forbidden status code.

Reminder: Country Filtering is applied recursively to a secured location. For example, an asset that is located in a subfolder of a secured directory will still be protected by Country Filtering.

Token-Based Authentication

If the requested content has been secured through Token-Based Authentication, then an edge server will look for a token value in the query string. If it finds one, then the token value will be decrypted according to the encryption key(s) specified for the HTTP Large platform. A check will then be performed to see whether the HTTP client (e.g., browser) meets the security requirements defined by the decrypted token value.

Note: Content can be secured by folder location or through HTTP Rules Engine. For more information, please refer to either the **Token-Based Authentication Administration Guide** or the **HTTP Rules Engine Administration Guide**.

By default, an HTTP client will receive a 403 Forbidden status code when any of the following is true for an asset protected by Token-Based Authentication:

- The CDN or edge CNAME URL did not include a token value in the query string.
- The specified token was invalid. This occurs for tokens generated using an invalid encryption key or an invalid parameter was used when generating an encryption key.
- The HTTP client did not meet the token requirements.

Tip: Instead of returning a 403 Forbidden status code for the cases mentioned above, you can take advantage of custom denial handling to redirect users or to report a different HTTP status code.

HTTP Rules Engine

HTTP Rules Engine is a powerful tool that can be used for a variety of purposes. One feature provided in this tool is "Deny Access (403)." This feature will return a 403 Forbidden status code to the HTTP client whenever a user-defined set of conditions are met. For information on how you can define the conditions (i.e., match options) that will be used to deny access to sensitive content, please refer to the **HTTP Rules Engine** guide.

Phase 3: Check for Cached Content

If the requested asset passes all of the above security requirements, then a check is performed to see whether the asset is currently cached on the POP handling the request. If an asset has been previously requested from the region covered by that POP and its content headers allowed it to be cached, then a cached asset may already be on that POP.

Check for Content Freshness

If a cached version of the requested asset is found on the POP handling the request, then a check for content freshness will be performed.

Note: The Origin Shield feature, which must be purchased separately, provides an intermediate cache layer between your origin server and our POPs. If you have enabled it on your customer origin, then please refer to the origin shield version of this procedure below.

Note: If the cached version of the requested asset contains a "Cache-Control: must-revalidate" header value, then our edge and origin shield servers will honor it. This means that they will always check the origin server for a newer version of a stale asset with this header value.

Check for content freshness (default):

1. An asset's time to live (TTL) will be calculated through its Cache-Control and Expires headers. If the asset is still fresh, then it will be immediately delivered to the client.
2. If the asset has exceeded its TTL, then the POP will check either a CDN or customer origin server for a new version.
    - If the cached version of the asset is still valid, then the response headers for the asset cached on the edge server will be updated to reflect a new TTL. Additionally, the cached version of the asset will be delivered to the client.
    - If the cached version of the asset is no longer current, then the new version of the asset will be retrieved from the origin server. After which, an edge server on the corresponding POP will deliver it to the client. Additionally, the asset will be cached for future retrieval from that POP.

Check for content freshness (Origin Shield):

1. An asset's time to live (TTL) will be calculated through its Cache-Control and Expires headers. If the asset is still fresh, then it will be immediately delivered to the client and no further steps will be performed.
2. If the request was directed at a customer origin that is protected by the Origin Shield feature, then an origin shield server will be checked for cached content.

- If a cached version of the asset is found, then it will be checked for freshness.
    - If the TTL is valid, then the following will take place:
        - The asset will be served to the client via an edge server corresponding to the POP closest to the client.
        - It will then be cached on that edge server.
    - If the TTL is expired, then the cached version on the origin shield server will be compared against the customer origin server.
        - If the same version of the asset is found on the customer origin server, then the following will take place:
            a. The headers on the origin shield server will be updated.
            b. The asset will be served to the client via an edge server corresponding to the POP closest to the client.
            c. It will then be cached on that edge server.
        - If a newer version is found, then the following will take place:
            a. An origin shield server will receive a new version of the asset.
            b. The origin shield server will forward it to an edge server corresponding to the POP closest to the client.
            c. It will then be cached on that origin shield server.
            d. The edge server will deliver it to the client.
            e. It will then be cached on that edge server.
- If cached content is not found, then the following will take place:
    i. The asset will be requested from the customer origin server and an origin shield server will receive it.
    ii. The origin shield server will forward it to an edge server corresponding to the POP closest to the client.
    iii. It will then be cached on that origin shield server.
    iv. The edge server will deliver it to the client.
    v. It will then be cached on that edge server.

Phase 4: Retrieval, Delivery, and Caching

If an asset is not currently cached on the POP handling the request or if the cached version of the asset is outdated, then a new version will need to be retrieved from either the CDN or customer origin server. The edge server on the respective POP will request the asset from the CDN or customer origin server.

Once it starts receiving the asset from the origin server, the edge server will immediately start delivering it to the client that requested it. Once the entire asset has been retrieved from the origin server, it will be cached on that POP. By default, the Cache-Control and Expires headers that are generated by the origin server are associated with that asset. If header information is missing for that asset, then the default max-age of that asset will be 7 days. This means that a cached version of that asset will be stored on the POP and it will be treated as fresh content for 7 days.

Note: HTTP Rules Engine, which must be purchased separately, can be used to customize how our CDN caches assets. For example, you can configure our CDN to cache an asset even if a user aborts his/her download. For more information, please refer to the **HTTP Rules Engine Administration Guide**.

Common HTTP Status Codes

A brief explanation is provided below for the common HTTP status codes that may be returned by a CDN or customer origin server.

Note: The following table describes the default HTTP status codes that will be reported by your HTTP client. Keep in mind that these status codes can be overridden by HTTP Rules Engine or Token-Based Authentication.

HTTP Status Code	Description
200 OK	An origin server returns this status code when it authorizes a request for an asset that is not currently cached on the POP serving the region from which the request originated. Unless the asset's headers specifically prohibit caching, the default behavior will cause the asset to be cached on that POP.
304 Not Modified	An origin server returns this status code when it authorizes a request for a stale asset that is cached on the POP serving the region from which the request originated and that asset has not been modified since it was cached on that POP. The POP will serve the currently cached version of the asset directly from the edge server to the requestor. The Cache-Control and Expires headers for that asset will be updated.
403 Forbidden	An origin server returns this status code when an unauthorized request is received. The requestor will be denied access to the requested asset. Additionally, a check will not be performed on the origin server to verify the freshness for the cached version of the requested asset. A request for an asset protected by Token-Based Authentication is considered unauthorized when it does not include a valid token or if the requestor does not meet the conditions specified in the included token.
404 Not Found	An origin server returns this status code when the requested asset was not found.
504 Gateway Timeout	An edge server returns this status code when it is unable to communicate with an origin server when trying to revalidate the requested asset.

Configuring an Origin Server

Overview

Before assets can be served through our CDN to your customers, they must first be retrieved from a web server. This type of server is known as an origin server, since it is the origin or source of the content that will be delivered to your users. A brief description is provided below for each type of origin server.

Origin Server Type	Description
Customer Origin Server	Our CDN service can be configured to recognize a web server that is external to our network (e.g., web hosting). The two main advantages of using a customer origin server are: • **Minimal Setup:** Setting up a customer origin server is as simple as creating a customer origin configuration from the MCC. All of your data can remain as it is currently structured on your web server. This can save you the amount of time it takes to upload and maintain that data on a separate server. • **Response Headers:** Our edge servers will forward the majority of response headers provided by your web server to your users. A benefit from this behavior is that you can define the cache instructions that will be assigned to your assets on your web server without having to purchase HTTP Rules Engine.
CDN Origin Server	Our CDN storage service allows you to upload content to a CDN web server dedicated to file storage. This type of server is known as a CDN origin server. Content in CDN storage can then be distributed to your users via our network. Our CDN origin servers are located within our CDN network which provides the following main benefits: • **Protection:** Our highly distributed network provides protection against network issues due to denial of service attacks or high traffic volume. This guarantees that the data will be available for distribution to your clients. • **Speed:** Data does not have to be retrieved from a location outside our network. This allows it to fully leverage our high-speed network to deliver content faster to your clients.

Customer Origin Server

Content stored or generated from a third-party web server (e.g., web hosting) can be delivered through our network. This requires that you define a customer origin configuration. This type of configuration maps one or more servers to a CDN URL.

Key information:

- The maximum number of customer origin configurations per platform is 100.
- Each configuration is specific to a platform.
- A directory name must be specified when defining a customer origin configuration. This directory name identifies your customer origin configuration in a CDN URL.
- Since CDN URLs are case-sensitive, make a note of the case used when defining the directory name.
- An edge CNAME configuration and a CNAME record may be used to generate a user-friendly URL known as an edge CNAME URL. For more information, please refer to the **Masking the URL of a CDN Origin Server (CNAME)** section.
- Define the set of servers that will fulfill HTTP requests.
    - If the SSL Traffic feature has been enabled on the platform in question, then you may also define a set of servers that will fulfill HTTPS requests.

2. A customer origin configuration consists of the following settings:
    - Origin Configuration
    - Directory Name
    - Domains/IP Addresses
    - HTTP Host Header
    - Load Balancing
    - Origin Shield

Origin Configuration

Domains associated with a customer origin configuration must be resolved to an IP address before a request can be served to it. The **Origin Configuration** option determines whether our servers will prefer to resolve a domain to an IPv4 or IPv6 address.

Setting	Description
Default	Indicates that domains should be resolved to IPv4 addresses only. **Note:** We reserve the right to change the behavior of this default setting at any time.
V6 Preferred Over V4	Indicates that our edge servers will resolve domains to IPv6 addresses whenever possible. If an IPv6 address for that domain does not exist, then the domain will be resolved to an IPv4 address.
V4 Preferred Over V6	Indicates that our edge servers will resolve domains to IPv4 addresses whenever possible. If an IPv4 address for that domain does not exist, then the domain will be resolved to an IPv6 address.
V4 Only	Indicates that domains should be resolved to IPv4 addresses only.
V6 Only	Indicates that domains should be resolved to IPv6 addresses only.

Directory Name

The **Directory Name** option defines how your customer origin configuration will be identified in a CDN URL. This name is specified as a part of the content access point (i.e., /80xxxx/**directory**). Our edge servers interpret the content access point to identify the starting location for your assets.

For example, if the primary purpose of your origin server is to serve images, then you might create a customer origin configuration whose **Directory Name** option is set to "images." A sample CDN URL that may be used to access assets on this customer origin server is provided below.

```
http://wpc.xxxx.edgecastcdn.net/80xxxx/images
```

Note: The term "*xxxx*" represents your CDN account number.

The above sample CDN URL points to the root folder on the server(s) associated with the "images" customer origin configuration. Append the desired relative path to the content that you would like to request. For example, the following sample CDN URL points to this location "http://www.myserver.com/photography/clientX" on your customer origin server:

```
http://wpc.xxxx.edgecastcdn.net/80xxxx/images/photography/clientX
```

Tip: If you would like to use a more user-friendly URL, then you should take advantage of an edge CNAME.

Note: CDN and edge CNAME URLs are case-sensitive.

Domain – IP Address Configuration

A customer origin configuration must point to one or more web servers. These web servers are defined through the **Hostname Configuration** section. The **HTTP Edge Protocol** and the **HTTPS Edge Protocol** options in that section define the set of servers that can fulfill HTTP or HTTPS requests, respectively.

Note: The ability to define a set of web servers that can fulfill HTTPS requests requires the SSL Traffic feature for the HTTP Large platform. This feature, in combination with an SSL certificate and an edge CNAME, allows the use of an HTTP Secure (HTTPS) URL when delivering content over our network. Please contact your CDN account manager to activate this feature.

Key information:

- Use one of the following formats to define a web server:

Format	Example
protocol://*domain*:*port*	http://www.mydomain.com:80
protocol://*IPv4Address*:*port*	http://10.10.10.255:80
protocol://[*IPv6Address*]:*port*	http://[1:2:3:4:5:6:7:8]:80

- An optimal configuration requires that the servers specified for each customer origin reside in relatively close vicinity to one another. If you would like to specify one or more servers that are located in a different geographic region, then we highly recommend that you create a separate customer origin configuration for those servers.
- In order to avoid DNS latency when loading content from a customer origin server, our edge servers proactively resolve each domain defined in a customer origin configuration.
- Our edge servers respect the DNS TTL defined for an origin server when resolving a domain.
- If our edge servers are unable to resolve a domain or if the domain's DNS TTL has expired, then they will reattempt to resolve it at regular intervals until it is resolved. Since this action may incur additional charges by your DNS provider, it is recommended that all domains associated with a customer origin configuration be registered in DNS with a reasonable TTL (e.g., 3600 seconds or more).
- Any combination of domains and IP addresses may be used to identify the set of origin servers that can honor requests for the customer origin being configured.

- The set of domains and IP addresses defined for either the **HTTP Edge Protocol** or the **HTTPS Edge Protocol** list must be assigned the same protocol (i.e., HTTP or HTTPS).
- Defining more than one server for either the **HTTP Edge Protocol** or the **HTTPS Edge Protocol** list will cause requests to be load balanced across those origin servers for that protocol.
- A maximum of 10 domains and/or IP addresses per protocol (i.e., HTTP or HTTPS) can be associated with a customer origin configuration.

Reminder: A domain must be resolved to an IP address before a request can be forwarded to it. The **Origin Configuration** option defines how domains will be resolved to an IP address.

HTTP Requests

The **HTTP Edge Protocol** option defines the set of web servers that can handle HTTP requests.

Key information:

- Make sure that all web servers defined for the **HTTP Edge Protocol** option use the same protocol (i.e., either HTTP or HTTPS).
- If a valid origin server has not been specified under the **HTTP Edge Protocol** option, a 404 – Not Found error message will be returned to the user when content is requested from your origin server using the HTTP protocol.

HTTPS Requests

The SSL Traffic feature enables an additional customer origin configuration option called "HTTPS Edge Protocol." This option functions in the same way as the **HTTP Edge Protocol** option, except that it determines how HTTPS requests are handled.

Key information:

- The HTTP protocol may be used to fulfill HTTPS requests, but this type of configuration would sacrifice end-to-end encryption of the request and response. If this describes your preferred configuration, please make sure that all servers are identified using the HTTP protocol.
- If a valid domain has not been specified under the **HTTPS Edge Protocol** option, a 404 – Not Found error message will be returned to the user when content is requested from your origin server using the HTTPS protocol.
- Base CDN and edge CNAME URLs can be viewed from the **SSL URL** section on the HTTP Large home page.

HTTP Host Header

HTTP 1.1 requires a Host header to be sent with each request. A Host header identifies the domain/IP address and port associated with a request. This is especially useful when there are multiple virtual domains hosted on a single physical server or load-balanced set of servers.

Each customer origin configuration allows a Host header value to be configured. It is recommended that you set this value to either one of the domains/IP addresses from the HTTP/HTTPS hostname lists, or else to an edge CNAME that references this customer origin server.

Note: By default, the **HTTP Host Header** option is set to the first domain or IP address defined under the **Hostname Configuration** section.

Note: If the **HTTP Host Header** option is set to blank, then the request URI determines the value for the Host request header.

Load Balancing

The domains/IP addresses associated with the **HTTP Edge Protocol** option are resolved into a list of IP addresses through DNS. A separate list is generated for the **HTTPS Edge Protocol** option. If multiple unique IP addresses are associated with either option, then the selected load balancing mode determines which customer origin server will handle the next request.

The available load balancing options are:

- **Round Robin:** This mode will balance requests between all of the servers listed for a particular protocol. In other words, it will send the first request to the first server on the list. The next request will be sent to the next server on the list.
- **Primary & Failover:** This mode indicates that the specified servers form an ordered failover list. In other words, all requests will be forwarded to the first server on the list. If that server is unavailable, then the request will be sent to the next server on the list. This process continues until a server is able to honor the request. A server is unavailable when a TCP connection is refused or if the connection times out. The order of the failover sequence can be controlled by rearranging entries through the up/down double arrow buttons.

Note: These load-balancing options are completely independent from any load balancing that may already exist on the customer origin server. For instance, traffic for a single IP address might be balanced across several physical servers.

Origin Shield

An origin shield allows you to place an additional buffer between a customer origin server and your clients. This buffer is useful for protecting your customer origin server from:

- Denial of service attacks
- Spikes in traffic

Note: The Origin Shield feature is only available after it has been activated on the HTTP Large platform.

How Does It Work?

The Origin Shield feature provides an intermediate caching layer between the customer origin server and an edge server. This provides the following functionality:

- Content requested from a customer origin server can be cached on our origin shield servers.
- If an edge server does not have a cached version of the requested content, then it will forward the request to an origin shield server. If the origin shield server contains fresh content, then it can respond with it.
- If an edge server has a stale version of the requested content, then it can revalidate the freshness of the content through the origin shield server.

This intermediate cache layer reduces the number of requests that are sent to the customer origin server. This results in reduced server and network load on the customer origin server.

Configuration

Protecting a customer origin through the use of the Origin Shield feature requires that you either specify a single or multiple origin shield locations. A single location is the recommended configuration. Each configuration method is described below.

Note: Each origin shield server is identified by the city and the three-letter abbreviation for the POP where it is located. The use of a POP abbreviation allows you to distinguish between multiple origin shield servers in the same city.

Single Origin Shield Location (Recommended)

The recommended configuration for reducing requests to your customer origin server is to define a single origin shield location. This can be achieved by selecting the **Single POP** option and then selecting the location closest to the server(s) associated with the customer origin. This type of configuration allows all edge servers to request content from the specified origin shield. If the origin shield does not have the requested asset, it will forward the request to the customer origin server. After which, it will cache the asset and serve it to the edge server that

requested it. The edge server will also cache the asset and deliver it to the client that requested it.

Multiple Origin Shield Locations

An alternative configuration is to define several origin shield locations for a single customer origin configuration. This can be accomplished by selecting the **Multiple POPs** option and then choosing how requests are handled for each of the following regions: North America: West Coast United States, North America: East Coast United States, Europe, and Asia. Choose one of the following options for each region:

- **Blank:** Leaving a region blank indicates that requests for this region will skip the origin shield server in the selected POP and attempt to retrieve it from the next closest origin shield server.
- **POP:** Selecting a POP activates origin shield for that region. Requests for this region will go through the selected origin shield. If the origin shield does not have the requested asset, it will request it from the customer origin server.
- **Bypass:** Selecting to bypass a region indicates that requests for this region will bypass the origin shield and go directly to the customer origin server. This type of configuration is the equivalent of turning origin shield off for a particular region.

Note: Origin shield locations in Asia are only available if the Global + Premium Asia delivery region has been activated on your account.

Customer Origin Management

A customer origin configuration is required to serve content from your servers to your users via our CDN. This section explains how to create, modify, and delete customer origin configurations.

Creating a Customer Origin Configuration

This section provides step-by-step instructions on how to create a customer origin configuration.

Key information:

- A customer origin configuration must point to a DNS record that has been fully propagated by your DNS provider. This may be checked through the use of the "dig" command-line tool, which is a DNS query tool that allows you to query your DNS provider for the DNS record that corresponds to your domain. When performing this dig, it is recommended that you use the "+trace" parameter. This will ensure that the DNS provider provides a direct response on the DNS record for the specified domain. The syntax for this command is provided below.

  ```
  dig domain +trace
  ```

  In the bottom section of the response, you should see an A record that points your domain to an IP address. This indicates that your DNS record has been fully propagated. An excerpt from a sample response is shown below.

  ```
  www.mycustomerorigin.com.    3600    IN      A       10.10.10.101
  mycustomerorigin.com.        3600    IN      NS      dns02.dnsauthority.com.
  mycustomerorigin.com.        3600    IN      NS      dns01.dnsauthority.com.
  ;; Received 100 bytes from 100.100.100.101#53(100.100.100.101) in 20 ms
  ```

- As a precautionary measure, it is recommended that you set the DNS TTL for your domain to a low value until you have confirmed that our edge servers can properly communicate with your origin server. A low DNS TTL reduces the amount of time that an improper DNS configuration will affect your origin server, while increasing the number of DNS queries sent to your DNS provider.
- The required options for a new customer origin are a folder name, a domain/IP address, and an HTTP host header. For your convenience, the HTTP host header is automatically populated when you add the first domain/IP address.
- Make sure that your origin server does not restrict access to the IP address blocks listed on the **Customer Origin** page.
- It may take up to an hour for your new customer origin configuration to take effect.

To create a customer origin configuration

1. Navigate to the **Customer Origin** page. Load this page by finding the **HTTP Large** menu and then selecting **Customer Origin**.
2. In the **Directory Name** option, type the name of the folder that will be associated with the desired customer origin server. This folder will become a part of the contact access point in the CDN URL (e.g., http://wpc.0001.edgecastcdn.net/800001/FolderName).
3. Perform one of the following:
    - If you would like customers to access the content of the customer origin server being configured through the HTTP protocol, then you should:
        i. Make sure that the **HTTP Edge Protocol** option is marked.
        ii. In the **Hostname or IP Address** option, you should type the domain or IP address of the customer origin server followed by a colon and the port through which communication will take place.
        iii. Click **Add**.
        iv. Repeat this step until you have finished adding all of the domains that will be associated with this customer origin for http:// requests.
    - If you would like customers to access the content of the customer origin server being configured only through the HTTPS protocol, then the **HTTP Edge Protocol** option should be cleared and you should proceed to the next step.
4. If you have purchased the SSL traffic feature for this platform, perform one of the following:
    - If you would like customers to access the content of the customer origin server being configured through the HTTPS protocol, then you should:
        i. Make sure that the **HTTPS Edge Protocol** option is marked.
        ii. In the **Hostname or IP Address** option, you should type the domain or IP address of the customer origin server followed by a colon and the port through which communication will take place.
        iii. Click **Add**.
        iv. Repeat this step until you have finished adding all of the domains that will be associated with this customer origin for https:// requests.
    - If you would like customers to access the content of the customer origin server being configured only through the HTTP protocol, then the **HTTPS Edge Protocol** option should be cleared and you should proceed to the next step.
5. Verify or set the **HTTP Host Header** option to one of the following:

- A domain specified in either step 3 or 4.
- An edge CNAME that points to a domain defined in either step 3 or 4.
- Blank. The request will determine the value assigned to the Host request header.

6. If you have purchased the Origin Shield feature, perform one of the following:
    - If you would like to use origin shield for the specified customer origin server(s), then you should:
        i. Make sure that the **Enable Origin Shield** option is marked.
        ii. Perform one of the following:
            - Set up a recommended origin shield configuration by selecting the **Single POP** option. You should then select the POP closest to your customer origin server(s) from the **ALL POPs** list.
            - Create a custom origin shield configuration by selecting the **Multiple POPs** option and then selecting the origin shield action that will take place for each region.
    - If you would like all requests for assets that have not been cached to be handled by your customer origin server, then you should make sure that the **Enable Origin Shield** option is cleared.
7. Click **Add** to save your customer origin configuration.

Modifying a Customer Origin Configuration

A customer origin configuration can be modified at any time by clicking the next to the desired customer origin. The configuration associated with that customer origin will appear. Simply make the desired changes and then click **Update** to apply them.

Note: If an edge CNAME points to a customer origin configuration, then you will not be allowed to modify the name of the folder associated with that customer origin configuration. If you would like to change the folder name, you will need to first delete the associated edge CNAME.

Note: It may take up to an hour for changes to your customer origin configuration to take effect.

Deleting a Customer Origin Configuration

A customer origin configuration can be deleted at any time by clicking the next to the desired customer origin. Once you have confirmed the deletion, it will be removed from the list.

Note: If an edge CNAME points to a customer origin configuration, then you will not be allowed to delete the associated customer origin configuration. If you would like to delete it, you will need to first delete the associated edge CNAME.

Note: It may take up to an hour for the deletion of your customer origin configuration to take effect.

Response Headers

When setting up your origin server, make sure that it adds either a Last-Modified (preferred) or an ETag header to each response. These response headers allow our edge servers to revalidate cached content. If either response tag is not found, then our edge servers will perform an unconditional GET request to the customer origin server.

Firewall Access

For the purpose of fulfilling requests, our edge servers require access to all servers associated with a customer origin configuration. Please ensure that your firewall allows access to all of the IP blocks listed in the **Whitelist IP Blocks** section of the **Customer Origin** page.

Tip: Export a list of the IPv4 and IPv6 blocks that should be whitelisted by clicking from the **Whitelist IP Blocks** header.

The **Whitelist IP Blocks** section contains a superset of IP addresses that includes:

- The IP blocks defined under the **The following CDN IPs can access your origin** section.
- The IP blocks for future POPs.

Note: Once the IP blocks defined under the **Whitelist IP Blocks** section have been whitelisted on your firewall, it is unnecessary to add the IP blocks defined under the **The following CDN IPs can access your origin** section.

Best Practices (Dynamic Application)

If your customer origin hosts a dynamic application, then it is highly recommended that you do not use a user's IP address to maintain a session instance. This type of configuration is unsupported, since the client does not connect directly to the customer origin server. Instead, the client connects to a server on our network, and then that server connects to a customer origin server. If you would like to maintain a session for your dynamic application, we recommend that you use a cookie to identify the session. For example, a cookie could keep track of a unique ID for each client's session.

CDN Origin Servers

Our CDN storage solution allows the storage of content on our storage servers (i.e., CDN origin servers). Content stored on our CDN storage solution can then be made available to clients either through:

- A system-defined URL (i.e., CDN URL)
- An edge CNAME URL

Note: A list of CDN and edge CNAME URLs that may be used to access content in our CDN storage solution may be viewed from the **URLs for CDN Origin** section of the HTTP Large home page.

Key information:

- Update and manage content on CDN storage using any of the following solutions:
  - SFTP protocol
  - rsync protocol
  - FTP protocol (insecure)
- Access content uploaded to CDN storage through the following base CDN URL:

  ```
  http://wpc.xxxx.edgecastcdn.net/00xxxx/
  ```
- Use an edge CNAME to provide a more user-friendly URL to your clients.

Note: For detailed information on how to upload content to CDN storage, please refer to the **CDN Storage - Data Upload** guide.

Masking the URL of a CDN Origin Server (CNAME)

The URL to the location on a CDN origin server where your assets reside can be masked through the use of a Canonical Name (CNAME) record and an edge CNAME. An edge CNAME informs our edge servers of the CNAME record that you would like to use, the domain used by that CNAME, and an optional path to which URLs that take advantage of the edge CNAME can be rewritten.

Important: Informing our edge servers that you would like to use a CNAME record will not update or set that CNAME record on your DNS server. Before you can take advantage of the desired CNAME, you will need to set a CNAME record on your DNS server.

Example: Instead of linking to assets using the following URL: http://wpc.0001.edgecastcdn.net/000001/Videos/Technical, you could create a CNAME record that points to that folder (e.g., technical.videos.com). You could then link to the same folder using this user-friendly URL: "http://technical.videos.com."

Note: You may also take advantage of an edge CNAME to mask the URL of your customer origin server.

When configuring our edge servers to recognize a CNAME, you should keep in mind all of the following:

- You should not specify the type of protocol that you will be using (e.g., http://, rtmp://, ftp://, etc.).
- All alphabetical characters in the CNAME should be specified in lower-case letters.
- You will need to set a CNAME record on your DNS server. The alias set on the DNS server should match the domain (e.g., wpc.*xxxx*.edgecastcdn.net) associated with the edge CNAME configuration. If a directory path has been specified, then our servers will rewrite the URL to point to the appropriate folder.
- Adding, modifying, or deleting an edge CNAME may take up to an hour to take effect.
- The EC360 Analytics Suite offers a variety of different ways to view CDN usage activity. However, if you would like to organize report data by edge CNAME, then you will need to enable the **Custom Reports** option on each desired edge CNAME. This will cause our servers to keep track of the amount of data transferred and hits for each selected edge CNAME. A icon will appear next to each edge CNAME for which reporting has been enabled.

Note: A custom report can be generated by finding the **Analytics** menu and then selecting **Custom Reports**. For more information, please refer to the **Custom Reports** chapter in the **EC360 Analytics Suite** guide.

To configure our edge servers to recognize your CNAME

1. Navigate to the **Edge CNAMEs** page. Load this page by finding the **HTTP Large** menu and then selecting **Edge CNAMEs**.
2. In the **New Edge Cname** option, type the name of the desired CNAME record. The CNAME should be specified in lower-case letters and should not include the protocol (i.e., http://).
3. Select whether the specified CNAME will point to a customer origin or CDN origin server.
4. By the **Points to** option, select the root location on the origin server to which the CNAME will be pointed. If you would like to indicate a specific folder, then you should type a forward slash (/) followed by the path to the desired folder.
5. In the **Custom Reports** option, choose whether to enable custom data logging for the edge CNAME being created.
6. Click **Add**.
7. Make sure that a CNAME record that points to the same domain has been registered on your DNS server. This CNAME record must match the name assigned to your edge CNAME.

To modify an edge CNAME

1. Navigate to the **Edge CNAMEs** page.
2. Click the next to the edge CNAME that you would like to modify.
3. Make the desired changes.
4. Click **Update** to save your changes.
5. If you have modified the **New Edge Cname** option, make sure to update the CNAME record registered on the DNS server.

To delete an edge CNAME

1. Navigate to the **Edge CNAMEs** page.
2. Click the next to the edge CNAME that you would like to delete.
3. When prompted, confirm the deletion of the selected edge CNAME.
4. Make sure to update or delete the corresponding CNAME record via a DNS service provider. Please refer to the **Setting up DNS for an Edge CNAME Deactivation** section for more information.

Note: HTTP Rules Engine allows you to match all requests that originate from a particular edge CNAME. However, an edge CNAME cannot be deleted if it is referenced by a rule. In order to delete the edge CNAME in question, please make sure to first modify or delete all match options that reference it.

Setting up DNS for an Edge CNAME Deactivation

A best practice for a deactivated edge CNAME configuration is to perform one of the following actions via your DNS service provider:

- Update the corresponding CNAME record to point away from our CDN service.
- Remove the corresponding CNAME record from the DNS zone.

This best practice is designed to reduce your risk exposure.

Edge CNAME Deactivation

An edge CNAME configuration may be deactivated due to any of the following actions:

- Edge CNAME configuration deletion
- Customer account deactivation
  A customer account will be deactivated under any of the following conditions:
  - A customer account was deleted.
  - A suspended customer account experiences an extended period of inactivity.
  - A trial account has expired.

Securing Your Content

Overview

There are three security tools provided to protect your content when handling HTTP and HTTPS requests. These tools are:

- **HTTP Secure (HTTPS):** This security mechanism uses Secure Socket Layer (SSL) to transmit encrypted data to ensure secure end-to-end data transmission.
- **Token-Based Authentication:** This security mechanism uses encrypted token values to ensure that the client requesting an asset is properly authorized.
- **Country Filtering:** This security mechanism prevents users from certain countries from accessing your data.

This section will explain how all three of these security mechanisms can be implemented.

Secure HTTP Requests (HTTPS)

If your account has been upgraded to use our Certificate Provisioning System, then you may provision a new DV or OV certificate through it. Otherwise, you need to request it via your CDN account manager.

Certificate Provisioning System

A default CDN configuration allows traffic to flow over HTTP. Serving traffic over HTTPS requires a TLS certificate on our network, updating your CDN configuration, and defining a CNAME record via your DNS service provider.

Important: Delivery over HTTPS requires TLS activation and TLS certificates. Contact your CDN account manager for more information.

TLS Certificate Support

Our CDN supports the use of any certificate derived from a Certificate Authority (CA) by allowing you to bring your own certificate to our network. Additionally, you may purchase a TLS certificate that supports multiple Subject Alternative Names (SAN). We support the following levels of validation for our certificates:

- **Domain Validation (DV):** This validation level only requires proof that you control the domain(s) associated with the certificate. Use a Domain Control Validation (DCV) method (i.e., email or DNS record) to prove your control over those domains.
- **Organization Validation (OV):** In addition to domain validation, OV requires a CA representative to contact/vet your organization and check whether the applicant may use the requested domain(s).
- **Extended Validation (EV):** EV provides the highest level of validation by requiring the CA to perform a rigorous audit to establish identity assurance. Browsers use visual cues to identify sites that leverage an EV certificate. These visual cues provide additional reassurance to your clients that they are accessing a secured site.

Quick Start

Setting up HTTPS delivery involves the following steps:

1. Activate TLS and purchase the desired TLS certificate(s). Contact your CDN account manager for more information.
2. Set up the desired TLS certificate(s).
    - **New DV or OV Certificate:** Set up a new DV or OV certificate via the Certificate Provisioning System.
    - **New EV Certificate:** Contact your account manager to set up a new EV certificate.
    - **Existing TLS Certificate:** Submit a previously purchased TLS certificate by providing the intermediate certificate, public key, and the private key via the **SSL Certificate Submission** page.
3. Enable TLS 1.3 (recommended) and/or 1.2 encryption on your web server(s).

    Important: A recommended best practice is to disable support for SSL/TLS versions 1.1 or older.

4. **Customer Origin Only:** Configure your customer origin configuration for either end-to-end or client to edge encryption.
5. Create an edge CNAME configuration that points to a Subject Alternative Name (SAN) defined in the certificate (e.g., common name). Repeat this step as needed.

6. Once the TLS certificate has been deployed to our network, wait 6 hours and then verify that the certificate is live.
7. Create a CNAME record via your DNS service provider that points a SAN to the certificate's target CNAME. Repeat this step as needed.

DV and OV Certificate Setup

Setting up a DV or OV certificate requires performing the following steps:

1. Submit a request for a TLS certificate.
2. Prove your control over each Subject Alternative Name (SAN) defined in the certificate via a Domain Control Validation (DCV) method (i.e., email, DNS TXT, or DNS CNAME).
3. **OV Certificate Only:** Validate your organization by following the directions provided by the Certificate Authority (CA) via phone or email.
4. Wait until the CA has validated your request. After which, your certificate will be issued and installed on our network. Additionally, a target CNAME will be generated for each delivery platform and associated with your account.

Reminder: Once a TLS certificate has been installed on our network, you will need to update your edge CNAME configuration and DNS zone before HTTPS traffic may flow through our network. If you plan on serving HTTPS traffic via your web server(s), then you will also need to update your customer origin configuration.

To set up a DV or OV certificate

1. Navigate to the Certificate Provisioning System page. Load this page by finding the **More** menu and then selecting **Certificate Provisioning** from the **Tools** menu.
2. Click **Add Certificate**. The **New TLS Certificate** wizard will appear.
3. From the **Certificate Label** option, type a unique name that will be assigned to this request for a TLS certificate.
4. Skip the **Common Name** field. This field is auto-populated by the **Subject Alternative Name (SAN)** option.
5. From the **Domain Control Validation (DCV) Method** option, choose how you will prove your control over each domain associated with this request.
6. From the **Subject Alternative Name (SAN)** option, click **Add Domain** and then type the desired SAN. Repeat this step as needed.
7. If you have specified multiple SANs, then verify the common name defined in the **Common Name** option. Assign a different common name by clicking the **Is Common Name?** option next to the desired domain.
8. From the **Validation Level** option, select the level of validation that will be performed by the CA.

- **Domain Validation (DV):** Proceed to the next step.
- **Organization Validation (OV):** Provide your contact and organization information in the form that appears in this section. The CA will validate your organization using the provided information.

  Tip: Fill out this form with care. Misspellings or incorrect information (e.g., unregistered abbreviations) will delay your request until those issues are corrected.

9. Click **Submit Request** to submit your certificate request to the CA.
10. Wait approximately 15 minutes for the CA to process your request.
11. Open your certificate request by clicking on it from the **Certificate Provisioning System** page. Verify that you are now on step **3. Domain Validation (DCV)**.
12. Validate your domain according to the DCV selected in the **Domain Control Validation (DCV) Method** option.
    - **Email:** Follow the directions for both sets of emails sent by the CA to:
      i. The registered owner(s) of the public domain as determined by the domain's WHOIS record.
      ii. admin@Domain, administrator@Domain, webmaster@Domain, hostmaster@Domain, and postmaster@Domain.
    - **DNS Text:** Perform the following steps:
      i. Click **Generate** for each SAN defined in your certificate request.
      ii. Create a TXT record for each SAN defined in your certificate request and set it to that SAN's token.
    - **DNS CNAME:** Perform the following steps:
      i. Click **Generate** for each SAN defined in your certificate request.
      ii. Create a CNAME record for each SAN defined in your certificate request with the following properties:
        - **Name:** Set it to a SAN-specific token.
        - **Value:** Set it to dcv.digicert.com.
13. Once the CA is able to validate your control over the domains defined in the certificate request, your request will proceed to step **4. Other Validations**. In this step, the CA performs additional validations including Organization Validation (OV).
14. **Organization Validation (OV) Only:** In addition to other validations performed by the CA, you will be required to follow the directions provided by DigiCert (CA) via an email to the contact email.

15. Once the CA has approved your request, they will issue your certificate. Our CDN service will then install it on our network and generate a target CNAME for each delivery platform.
16. Wait 6 hours and then verify that the certificate is live.
    i. Dig the certificate's target CNAME. Note the IP address returned by the dig tool.
    ii. Update your hosts file to point the above IP address to a SAN (e.g., common name).
    iii. Point your browser to:
    https://SAN/
    iv. View the certificate by clicking on the certificate icon that appears in the browser's address bar. Verify that the common name matches the one that you requested.

> **Note:** If the browser returns an error, then the certificate has not been fully deployed. Please wait a few more minutes and then retry.

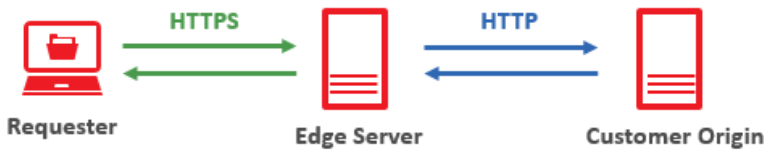
Customer Origin Configuration

> **Note:** This section only applies if you would like to serve HTTPS traffic from your own web servers. This section is inapplicable when setting up HTTPS delivery for CDN storage or Azure block blob storage.

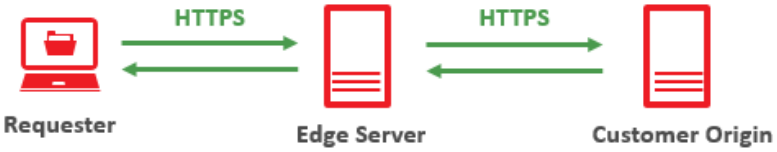
A customer origin configuration determines the scheme for communication between:

- The client and the edge of our network.
- The edge of our network and your web server(s).

A prerequisite for delivery over HTTPS requires your customer origin to use the HTTPS scheme when serving content to your clients. This is known as client to edge encryption.



You may configure your customer origin to also use the HTTPS scheme when communicating with your web server(s). This is known as end-to-end encryption.



Important: Setting up HTTPS support on your customer origin configuration varies according to whether you are using customer origin groups. Customer origin groups is a new capability. Setup instructions for both customer origin groups and legacy customer origin configurations are provided below.

To update a customer origin group to use the HTTPS scheme

1. Navigate to the **Origins** page. Load this page by finding the **HTTP Large** menu and then selecting **Customer Origin**.
2. Click on the desired customer origin group to expand it.

   Note: Content delivery over HTTPS requires an edge CNAME configuration that points to this customer origin group.

3. Update origin entries to use the HTTPS scheme by performing the following steps:

   i. From the **HTTP - Edge Protocol** section, click › next to the desired origin entry.

   ii. From the **Protocol Type** option, set it to **HTTPS Only**.

   iii. Click **Save**.

   iv. Repeat the above steps as needed.

   Note: The above configuration configures your origin entries to use end-to-end encryption. Alternatively, you may configure an origin entry to only encrypt traffic between the client and our network or to only encrypt traffic when the client submits a HTTPS request.

4. Review your customer origin group's TLS configuration by performing the following steps:

   i. Click › next to the desired origin entry.

   ii. From the **Group Settings** section, click **Origin TLS**.

   iii. Review and adjust how our edge servers perform TLS verification with your origin servers.

   iv. Click **Save**.

To update your legacy customer origin configuration to use the HTTPS scheme

1. Open the desired customer origin configuration.

   Note: Content delivery over HTTPS requires an edge CNAME configuration that points to this customer origin.

2. Optional. Disable the use of the HTTP scheme for client to edge communication by clearing the **HTTP Edge Protocol** option.
3. Enable the **HTTPS Edge Protocol** option.
4. Configure this option using one of the following methods:

   - **End-to-End Encryption:** Specify one or more HTTPS hostnames/IP addresses under the **HTTPS Edge Protocol** option.
     Sample Configuration:
     https://video.mydomain.com
     https://101.10.20.30
   - **Client to Edge Encryption:** Specify HTTP hostnames/IP addresses under the **HTTPS Edge Protocol** option.
     Sample Configuration:
     http://video.mydomain.com
     http://101.10.20.30

5. Click **Update** to save your changes.

Edge CNAME Configuration

Create or update an edge CNAME configuration for each Subject Alternative Name (SAN) defined in the certificate (e.g., common name).

To set up an edge CNAME for use with a TLS certificate

1. Navigate to the **Edge CNAMEs** page corresponding to the desired platform.
2. In the **New Edge Cname** option, type the desired SAN.

   Reminder: This hostname should be specified in lower-case letters and should not include a protocol (i.e., http://).

3. In the **Points To** option, select whether the edge CNAME will point to a customer origin or CDN origin server.
4. In the **Origin Directory** option, select one of the following:
   - **CDN Origin:** The selected origin directory (i.e., /00) points to CDN storage.
   - **Customer Origin:** Select the desired customer origin configuration (/80/*customerorigin*).

   Reminder: If you point it to a customer origin, then please verify that it has been updated to leverage HTTPS.

5. Click **Add**.

Reminder: Traffic may only be delivered via this edge CNAME configuration once your DNS has been updated. Delivery over HTTPS requires a CNAME record that points a SAN to the certificate's target CNAME.

Legacy HTTPS

By default, CDN traffic may flow over HTTP. Adding support for HTTP Secure (HTTPS) requires that your CDN account manager perform the following steps on your behalf:

- The SSL traffic feature must be activated on the desired platform.
- An SSL certificate that identifies the hostname over which secure CDN traffic will flow must be installed on our network.

Request support for either of the following types of certificates:

- **Hosted SAN SSL Certificate:** This is a shared SAN certificate that has an Edgecast-owned common name. The Subject Alternative Names associated with this certificate defines the domains for several customers.
- **Custom SSL Certificate:** This is an umbrella term for any type of certificate that is dedicated to your organization. This type of certificate can define a single domain,

multiple domains, or a wildcard domain (validates unlimited subdomains). Finally, Extended Validation (EV) status can be granted to this type of certificate. An EV SSL certificate provides additional visual reassurance to your clients that they are accessing a secured site.

Note: A previously purchased SSL certificate can be leveraged for CDN usage. This process will require that we install your SSL certificate on our servers. For additional details and pricing information, please contact your CDN account manager.

Setting up HTTPS Support

Support for HTTPS traffic requires that your CDN account meet all of the following requirements:

- The SSL traffic feature for this platform must be enabled on your account.
- An SSL certificate that references the desired domain must be installed on our network.
- An edge CNAME should point the hostname referenced in the SSL certificate to a CDN or a customer origin server.

Upon requesting HTTPS support, your CDN account manager will enable the SSL Traffic feature for the platform in question and request its deployment from CDN personnel. Perform the following steps while awaiting SSL certificate activation:

1. Create an edge CNAME named after the hostname defined for the requested SSL certificate.
2. If the above edge CNAME points to a customer origin configuration, make sure that the **HTTPS Edge Protocol** option has been enabled on that customer origin configuration.
    - **End-to-End Encryption:** Set up secure communication from a client to our network to your customer origin server by specifying one or more HTTPS hostnames/IP addresses (e.g., https://video.mydomain.com) under the **HTTPS Edge Protocol** option.
    - **Client to Edge Encryption:** Our network can be configured to communicate over HTTPS with your clients, while transmitting data over HTTP between our network and your customer origin server. This can be accomplished by specifying HTTP hostnames/IP addresses (e.g., http://video.mydomain.com) under the **HTTPS Edge Protocol** option.
3. Create a CNAME on the DNS server that points the edge CNAME's hostname to the hostname provided by your CDN account manager.

Reminder: Edge CNAME configuration changes typically take effect in 1 hour.

Token-Based Authentication

Token-Based Authentication provides security for assets accessed through our content delivery network. This feature allows you to protect your content by the country, URL, IP address, protocol, or the referrer that linked to your asset. Additionally, you can protect your content by only allowing it to be available for a certain amount of time. Regardless of how you decide to protect your content, only authorized users that provide the appropriate token for the requested asset will be able to access your content. For detailed information on how you can configure Token-Based Authentication, please refer to the **Token-Based Authentication Administration Guide**, which can be downloaded either from the MCC home page or the **Token Auth** page.

Country Filtering

The default configuration for the HTTP Large platform allows your content to be accessed from all countries. However, you can prevent users from certain countries from accessing your content. One way of accomplishing this is by configuring Token-Based Authentication to secure content by country. This type of configuration is discussed in the **Token-Based Authentication Administration Guide**. An easier implementation of authenticating by country is to use the Country Filtering feature. This section will explain how to configure this platform to only allow or block users by country.

Note: It may take up to an hour for changes to your country filtering configuration to take effect.

Administering Country Filtering Configurations

When configuring a country filter, you must specify the relative path to the location to which users will be allowed or denied access. The starting point for this relative path is defined below:

URL Type	Relative Path (Starting Point)
CDN URL	Specify a relative path that starts directly after the content access point (e.g., /000001, /200001, or /800001). **Sample URL:** http://wpc.0001.edgecastcdn.net/800001/customerorigin/videos/fly.flv In the above sample URL, the gray text indicates what should be excluded when securing a location. This sample request can be secured by any of the following configurations: • / • /customerorigin • /customerorigin/videos
Edge CNAME URL (CDN Origin)	Specify a relative path that starts directly after the domain. **Sample URL:** http://www.domain.com/presentations/sales/businessplan.ppt In the above sample URL, the gray text indicates what should be excluded when securing a location. This sample request can be secured by any of the following configurations: • / • /presentations • /presentations/sales
Edge CNAME URL (Customer Origin)	Specify a relative path that starts with the name of the customer origin configuration referenced by the edge CNAME URL. **Sample edge CNAME URL:** http://www.domain.com/Photos/Store.jpg Our edge servers will re-write the edge CNAME URL requested by the client (above) with the following CDN URL: http://wpc.0001.edgecastcdn.net/800001/customerorigin/Photos/Store.jpg The above CDN URL is used to determine the starting point for the relative path that should be secured. The gray text in the above CDN URL indicates what should be excluded when securing a location. This sample request can be secured by any of the following configurations: • / • /customerorigin • /customerorigin/Photos

Key information:

- The specified relative path is compared against all CDN and edge CNAME URLs that request content from this platform.
- The path to a protected folder always starts with a forward slash (/).
- The path to a protected folder may include a trailing forward slash (/).
- It may take up to an hour before a new location is fully protected.
- This feature does not support wildcard characters (e.g., *).
- The country filtering configuration associated with the relative path will be applied recursively to that path.
- Only a single country filter can be applied to the same relative path. In other words, you cannot create multiple country filters that point to the same relative path. However, a folder may have multiple country filters. This is due to the recursive nature of country filters. In other words, a subfolder of a previously configured folder can be assigned a different country filter.

Once a relative path has been specified, choose whether to allow or block users from the specified countries.

- **Allow:** Only users from the specified countries will be allowed access to assets requested from that recursive path.
- **Deny:** Users from the specified countries will be denied access to assets requested from that recursive path. If no other country filtering options have been configured for that location, then all other users will be allowed access.

Important: The "EU" and "AP" country codes do not allow or block all requests from those regions. Rather, those country codes are applied to requests that originate from IP addresses that are spread out over a region instead of a country. If you would like to allow or block all requests from a particular region, then you would need to select all countries in that region and the region-specific code (i.e., EU or AP).

To configure country filtering options

1. Navigate to the **Country Filtering** page. Load this page by finding the **HTTP Large** menu and then selecting **Country Filtering**.
2. Click **Add Filter** to launch the **Country Filtering** wizard.
3. Specify the path to the folder (e.g., /My_Folder) to which country filtering options will be applied. Keep in mind that this path applies to both CDN and customer origin servers.
4. Select whether to **Block** or **Allow** user access to the folder specified in the previous step.
5. Type the name of the country for which access will be allowed or denied and then press ENTER.
6. Repeat the previous step as needed.
7. Click **Save**.

To modify country filtering options

1. Navigate to the **Country Filtering** page. Load this page by finding the **HTTP Large** menu and then selecting **Country Filtering**.
2. Click the next to the desired country filtering configuration. The **Country Filter** wizard will display the settings for the selected country filter configuration.
3. Modify the desired settings.
4. Click **Save**.

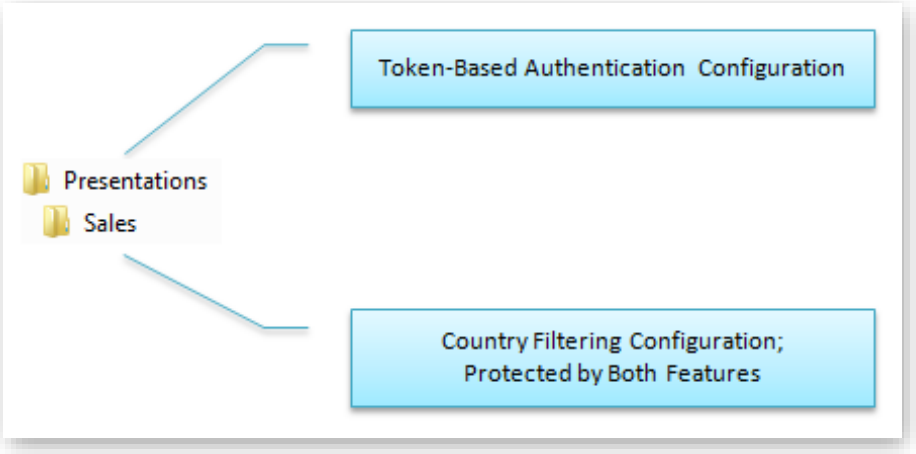
To delete country filtering options

1. Navigate to the **Country Filtering** page. Load this page by finding the **HTTP Large** menu and then selecting **Country Filtering**.
2. Click the next to the desired country filtering configuration. The **Country Filtering** wizard will display the settings for the selected country filter configuration.
3. Click **Delete**.
4. Click **I Understand, please delete this filter**.

Protecting Assets Using Both Country Filtering and Token-Based Authentication

A Country Filtering and a Token-Based Authentication configuration cannot be applied to the same folder. However, since both of these features are applied recursively, you may apply protection on a hierarchical basis. This can be seen in the following illustration.

Note: This same principle can be used to apply multiple Country Filtering configurations to a particular folder.



In the above example, a Token-Based Authentication configuration has been applied to a folder called "Presentations." The "Presentations" folder contains a folder called "Sales." A Country Filtering configuration has been applied to this subfolder. The "Presentations" folder will be protected by Token-Based Authentication, while the "Sales" folder will be protected by both Token-Based Authentication and Country Filtering.

An asset residing in a folder that is protected by both Token-Based Authentication and Country Filtering cannot be downloaded unless the requirements for both security mechanisms are met. In the above example, content stored in the "Sales" folder cannot be downloaded unless a valid token is provided and the Country Filtering configuration does not prevent users in your country from accessing it.

Cache Management

Default Cache Management

There are two types of caching that can take place when an asset is routed through our CDN, which are edge server and user agent (e.g., web browser) caching. Edge server caching determines whether the requested asset will be cached for a particular POP and for how long. This type of caching determines when an edge server on that POP will check for a new version from the origin server. User agent caching determines whether the requested asset will be cached on a web browser and for how long. By default, this type of caching determines when a user agent (e.g., browser) will revalidate the cached asset with the edge server.

Note: Keep in mind that clearing a web browser's cache will override user agent caching.

If an asset's headers do not prohibit caching, then the default behavior is to cache it on our POPs and on each user agent (e.g., web browser) that requests it according to the asset's Cache-Control and Expires header information. If an asset does not have header information, then it will be assigned a TTL of 7 days (i.e., Cache-Control: max-age=604800) when it is cached on our edge servers and by the user agents that requested it. You can override this default behavior by manually caching and purging assets from our POPs or through the use of HTTP Rules Engine.

Once the asset has been cached on a POP, all future requests that originate from the region served by that POP will be served directly from that POP while the asset is fresh. Once the asset's TTL has expired, then an edge server from that POP will revalidate the asset with the origin server. If the asset has not been modified, then the header information for the cached asset on the edge server and the user agent that requested it will be updated. Otherwise, a new version of the asset will be retrieved and cached on the edge server and the user agent.

When defining a default caching policy, keep the following items in mind:

- By default, our CDN honors the s-maxage, max-age, and Expires header assigned to an asset. An asset's TTL is determined by those directives and the Expires header in that order. Keep in mind that s-maxage only affects the TTL for edge server caching.
- If an asset contains an Expires header that has been assigned an invalid value, including zero, then the Expires header will be ignored for that asset.
- If an asset has been assigned both a no-cache and either an s-maxage or max-age directive, then the "Cache-Control: no-cache" directive will take precedence. On a related note, if an asset has been assigned "Pragma: no-cache", then it is treated as a "Cache-Control: no-cache" directive.

- If an edge server successfully revalidates a cached version of an asset, then the expiration time will automatically get updated. This means that the cached version of the asset will be assigned new Cache-Control and Expires headers based on the response returned from the origin server. If these headers have not been defined, then the asset will be assigned a default max-age of 7 days from the time it was successfully revalidated. Please refer to your web server's documentation for more information on how to configure these settings.

Caching & Incomplete Downloads (Complete Cache Fill)

If a client requests an asset that has not been previously cached on the corresponding POP, then an edge server will retrieve it from the origin server. The entire asset will then be cached on that edge server provided that its headers do not prohibit caching, regardless of whether the client performed an HTTP range request or aborted their download. By automatically retrieving and caching the entire asset, future requests for that asset will not require an edge server on that POP to download it again from your origin server while the asset is still fresh. This functionality reduces the load on the origin server and allows future requests to be served from cache. This ensures the speedy delivery of your content to your customers.

Loading Assets

Content can be loaded on to our network. A request to load content will cache the content in question across our entire network or within specific regions. This concept is also known as pre-caching.

The benefits of loading content are:

- Reduces server and network load on your customer origin server.
- Ensures optimal content delivery to your users.

Tip: Loading content is most effective when it deals with a large event or content that becomes simultaneously available to a large volume of users (e.g., a new movie release or a software update).

Key information:

- By default, content is cached as it is requested. This means that the first and/or second request from each region may take a little bit longer, since our edge servers will not have the content in cache and therefore will need to forward the request to the origin server. Loading content avoids this first/second hit latency.
- A request to load content is specific to the following criteria:
    - **Directory:** A URL must be specified when loading content. This URL identifies the directory where the content to be loaded is stored.
        - Loading is not URL-specific. In other words, it affects all CDN and edge CNAME URLs that point to the same asset.
    - **Region:** Loading by region is an advanced option that should only be used to address specialized cache management needs. It allows the selection of one or more regions when loading content. The asset in question will only be loaded to the POPs in the selected region(s).
        - By default, loading is applied to the entire network (i.e., all regions).
- An asset only needs to be loaded a single time per directory, region, and protocol.
- Each load request needs to be processed. Therefore, a load request will be placed in a queue. The Purge/Load page provides the means to keep track of when a request was made and when it was completed.
- There is a default limit of 50 concurrent load requests at any given time.

To load content

1. Navigate to the **Purge/Load** page.
2. From the **URL** option, which can be found under the **Load To Edge** section, select the base URL where the desired content can be found.
3. Directly below the **URL** option, type the relative path to the asset that will be loaded. Since a forward slash (/) has been added for your convenience, make sure that this relative path starts after the forward slash.
    - The base URL for this relative path is the URL selected in the previous step.
    - Make sure to include the file name extension when specifying the file name.
4. Click **Load Content**.

To perform a bulk load

1. Navigate to the **Purge/Load** page.
2. Click **Bulk Load**, which can be found under the **Load To Edge** section, to display the **URLs** option.
3. In the **URLs** option, type the CDN or edge CNAME URL that points to the asset(s) that will be loaded.
4. Add another load request by pressing ENTER and then specifying the desired CDN or edge CNAME URL. Repeat this step as needed.
5. From the **Regions** option, perform one of the following steps:
    - **Load to All Regions (Recommended):** Select **All**.
    - **Load to Specific Regions:**
        i. Select **Custom**.
        ii. Mark the regions where content should be loaded.
        iii. Clear all other regions.
6. Click **Load Content**.

Note: Each CDN or edge CNAME URL specified in the URLs option must be placed on a separate line. This can be accomplished by using a carriage return to delimit each CDN or edge CNAME URL. Using any other type of character (e.g., a comma) as a delimiter may prevent the asset from being loaded.

Automatically Loading Assets through the Web Services REST API

Content may be automatically loaded via our Web Services REST API. For more information, please refer to our REST API Help Center.

Reminder: Although the Web Services REST API allows you to quickly submit multiple load requests, the default limit of 50 concurrent load requests is still applicable.

Purging Assets

Cached content can be purged from:

- Our entire network (i.e., all POPs).
- Specific regions (i.e., subset of POPs).

The purpose of purging content is to force the CDN to request a new version of the content in question from an origin server. This ensures that the latest version of the content in question is delivered to your clients.

Note: Purging does not delete content from the origin server. A file management tool (e.g., SFTP or rsync) may be used to delete content from an origin server.

Note: There is a default limit of 50 concurrent purge requests at any given time.

Automated Purging Through the Web Services REST API

Content may be automatically purged via our Web Services REST API. For more information, please refer to our REST API Help Center.

Reminder: Although the Web Services REST API allows you to quickly submit multiple purge requests, the default limit of 50 concurrent purge requests is still applicable.

Manually Purging Assets

Cached assets can be purged from our edge servers through the **Purge/Load** page. Load this page by finding the **HTTP Large** menu and then selecting **Purge/Load**.

Keep the following information in mind when purging content:

- A request to purge content is specific to the following criteria:
    - **Directory:** A URL must be specified when purging content. This URL identifies the directory where the content to be purged is stored.
        - Purging is not URL-specific. In other words, it affects all CDN and edge CNAME URLs that point to the same asset.
    - **Region:** Purging by region is an advanced option that should only be used to address specialized cache management needs. It allows the selection of one or more regions when purging content. The asset in question will only be purged from the selected region(s).
        - By default, purging is applied to the entire network (i.e., all regions).
- Purges are protocol-independent (i.e., HTTP or HTTPS).
- An asset only needs to be purged a single time per directory and region.
- Each purge request needs to be applied to edge servers throughout our network. As a result, each request will be placed in a queue. The **Purge/Load** page provides the means to keep track of when a request was made and when it was completed.
- A URL containing a dollar sign symbol (i.e., $) cannot be added to the purge queue. The dollar sign symbol must be encoded as "%24" before such a URL can be added to the queue.
- Use purge syntax to determine the scope of the purge request. You can choose to purge assets individually, by folder, or recursively. Detailed information on how to perform each type of purge is described in the corresponding sections below.
- If you have assets that are updated frequently, then you should consider using a naming convention that assigns unique names to your new assets. This will ensure that future requests will point to the new assets without the need of manually purging the old assets.
- An alternative to changing your file naming convention for assets that are updated frequently is to use the **URLs** option. This option allows you to specify a list of assets that will be placed into the purge queue. If you keep a record of frequently purged assets, then you can copy and paste them into this option.
- There is a default limit of 50 concurrent purge requests at any given time.

To manually purge assets

1. Navigate to the **Purge/Load** page. Load this page by finding the **HTTP Large** menu and then selecting **Purge/Load**.
2. From the **URL** option, which can be found under the **Purge From Edge** section, select the base URL where the desired content can be found.
3. Directly below the **URL** option, type the path to the folder containing the asset(s) to be purged. Since a forward slash (/) has been added for your convenience, make sure that this relative path starts after the forward slash.
4. Perform one of the following:

Purge	Procedure	Sample Syntax
Asset	Append the name of the asset that you would like to purge to the relative path specified in step 3.	Sales/Gala_01.flv
Set of Assets	Determine the naming convention and/or file type that will be used to identify the assets that will be purged. Append that pattern to the relative path specified in step 3.	Sales/Gala*.fl*
Folder	Append an asterisk, a period, and another asterisk (i.e., /*.*) to the relative path specified in step 3.	Sales/*.*
Folder (Recursively)	Append an asterisk (i.e., /*) to the relative path specified in step 3.	Sales/*

5. From the **Regions** option, perform one of the following steps:
    - **Purge All Regions (Recommended):** Select **All**.
    - **Purge Specific Regions:**
        i. Select **Custom**.
        ii. Mark the regions that should be purged.
        iii. Clear all other regions.
6. Click **Purge Content**.

To perform a bulk purge

1. Navigate to the **Purge/Load** page. Load this page by finding the **HTTP Large** menu and then selecting **Purge/Load**.
2. Click **Bulk Purge**, which can be found under the **Purge From Edge** section, to display the **URLs** option.
3. In the **URLs** option, type the CDN or edge CNAME URL that points to the asset(s) that will be purged.
4. Add another purge request by pressing ENTER and then specifying the desired CDN or edge CNAME URL. Repeat this step as needed.
5. From the **Regions** option, perform one of the following steps:
    - **Purge All Regions (Recommended):** Select **All**.
    - **Purge Specific Regions:**
        1. Select **Custom**.
        2. Mark the regions that should be purged.
        3. Clear all other regions.
6. Click **Purge Content**.

Note: Each CDN or edge CNAME URL specified in the **URLs** option must be placed on a separate line. This can be accomplished by using a carriage return to delimit each CDN or edge CNAME URL. Using any other type of character (e.g., a comma) as a delimiter may prevent the asset from being purged.

Purging an Individual Asset

An individual asset can be purged by specifying a CDN or an edge CNAME URL that points to it. Keep in mind that this type of purge request will also purge all query string variations of the specified asset at that location. Sample URLs are provided below.

- **CDN URL:** http://wpc.0001.edgecastcdn.net/000001/resources/homepage.html
- **Edge CNAME URL:** http://cdn.sampleurl.net/resources/homepage.html

Purging a Set of Assets

A pattern can be defined through the use of asterisks to determine the assets that will be purged from the specified directory. Each asterisk represents one or more characters. If you would like to purge a particular file type, then you can simply append an asterisk, a period, and the desired filename extension to the purge URL (e.g., http://cdn.sampleurl.net/*.html).

A list of sample purges is provided below.

Example	Description
http://cdn.sampleurl.net/folder01/*.*	This sample purge request will purge the entire contents of the directory called "folder01." This purge request will not be performed recursively.
http://cdn.sampleurl.net/folder01/*.css	This sample purge request will purge all CSS assets from the directory called "folder01."
http://cdn.sampleurl.net/folder01/a*.ht*	This sample purge request will purge all assets that start with the letter "a" and whose filename extension starts with "ht" (e.g., activity.html).

An asterisk (*) can either be used as a wildcard in a filename pattern or to recursively purge a folder's content. However, it cannot be used as a wildcard when specifying a directory. Additionally, you should keep in mind a recursive purge is always performed for purge requests that end with a forward slash followed by an asterisk (/*). If you would like to only purge a folder's content, append /*.* to the desired purge directory.

Recursive Purging

We have already seen how to purge the contents of a folder through the use of wildcards (i.e., *.*). If you would prefer to purge a directory and all of its subfolders, then you should specify a purge request for that directory and append an asterisk. A sample purge request is provided below.

- http://cdn.sampleurl.net/folder/*

Reminder: An asterisk cannot be used to specify a folder pattern. It can only be used as specified above or when defining a pattern that will be used to define the set of assets that will be purged (e.g., /*.css).

Purging Query String Variations of a Cached Asset

If your file naming structure is based on query strings, you can customize how query string variations of that asset are handled. Several examples on how you can handle query string variations of an asset are provided below.

Sample Purge Request	Description
http://cdn.sampleurl.net/folder/sampleasset.htm	Simply specifying a filename will purge the original asset and all of its query string variations.
http://cdn.sampleurl.net/folder/sampleasset.htm?	A filename followed by a question mark (?) indicates that only the specified asset will be purged. All query string variations of that asset will not be purged.
http://cdn.sampleurl.net/folder/sampleasset.htm?*Query*	A filename followed by a question mark and a query string (?SampleQuery) indicates that only the specified asset with that particular query string will be purged. The original asset and all other query string variations of that asset will not be purged.

Purging a Directory's Default Cached Asset

The default asset cached for a particular directory can be purged through the use of the syntax described below.

Sample Purge Request	Description
http://cdn.sampleurl.net/folder/	The default cached asset for a directory, along with all of its query string variations, can be purged by specifying a CDN or edge CNAME URL to the desired directory.
http://cdn.sampleurl.net/folder/?	The default cached asset for a directory, excluding query string variations, can be purged by specifying a CDN or edge CNAME URL to the desired directory and appending a question mark.
http://cdn.sampleurl.net/folder/?*Query*	A query string variation of the default cached asset can be purged by specifying a CDN or edge CNAME URL to the desired directory and appending the desired query string.

Note: When purging the default asset that was cached for a particular folder, keep in mind that there are certain conditions that may cause an asset other than the default asset for that folder to be purged. For example, specifying "/Folder/" may not be the same as specifying "/folder/index.html" even if "index.html" is the default asset for that folder. A couple of situations under which this would occur are when the specified URL is redirected or if URL rewriting is used.

Purge and Load History

A log of the most recent purge and load requests is provided on the **Purge/Load** page. Purge and load history indicates:

- The CDN URL to the asset that was purged/loaded (**Path**).
- The set of regions to which the purge/load request was applied.
- When the request was made (**Created**).
- The date/time when the action was completed (**Completed**). If the **Completed** column is blank, then the request is still being processed.

Tip: Purge/load history is paginated. Leverage the navigation bar at the bottom left of the page to navigate between pages.

Note: Purge and load history is only displayed for the last 200 requests within the last 7 days. Requests older than a week will not be displayed. Additionally, once this limit has been reached, old purge or load requests will be cleared to make way for new requests.

Note: Purge/load history is reported using the GMT time zone.

Query String Caching

Requests that contain a query string can be cached differently. You can choose to cache these types of assets in one of three different ways. Each cache method is explained below.

Cache Method	Description
Standard-cache	This is the default mode for handling URLs that contain query strings (e.g., Asset.txt?Data=True). This mode will ignore query strings in the URL. It will only cache a single asset per URL, regardless of the presence of query strings. However, if there is a cache miss or an expired cache asset needs to be revalidated, then the full CDN URL, including query strings, will be passed to the customer origin server.
No-cache	This mode prevents requests containing query strings from being cached. All requests that contain query strings will be forwarded to the customer origin server.
Unique-cache	This mode will cache an asset for each request made with a unique URL. A unique cache asset will be created by even the slightest variation in the query string. It is recommended that you enable the query string logging feature when taking advantage of unique-cache mode. Additionally, it is highly recommended that this mode be avoided when requests will contain query strings that will result in a very low cache hit ratio. For examplefilter, information such as session ID or user name would create such a scenario.

Token-Based Authentication and Query String Caching

By default, no-cache or unique-cache mode cannot be used with Token-Based Authentication. This means that if you have protected at least one folder with Token-Based Authentication, then the only cache method that will be available from the MCC is standard-cache. Likewise, if you have configured query string caching to use no-cache or unique-cache mode, then you will not be allowed to protect a folder with Token-Based Authentication.

If you would like to protect your assets using Token-Based Authentication and you have other assets that require query string caching, then a custom configuration will have to be implemented for your account. This custom configuration will disable query string caching on folders that have been protected by Token-Based Authentication. If you would like such a custom configuration, then you should contact your CDN account manager.

Note: A custom configuration, such as the one described above, will cause the options that appear on the **Query-String Caching** page to be disabled.

Query String Logging

Query string logging determines whether the query string portion of a CDN or edge CNAME URL will be recorded in our raw access logs. Raw access logs are the log files from which we generate our reports. Keeping track of query string information is important when you have enabled unique-caching mode, which caches assets according to query string parameters. By keeping track of query string information, you can view statistics on each cached asset.

Asset Compression

Compression allows you to save on bandwidth costs and reduces the amount of time that it takes for your users to download your content. An asset can be compressed either by an origin server or an edge server. Both compression options are described below.

Customer Origin Server Compression

An origin server may compress an asset before delivering it to an edge server.

Setup

Origin server compression requires the following setup:

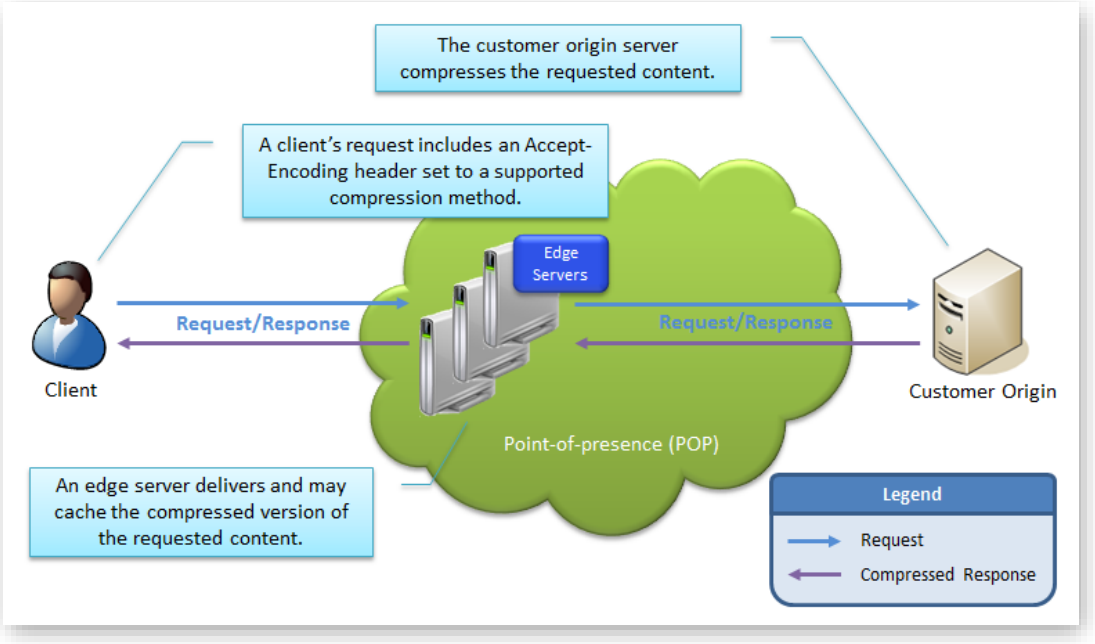
1. The request must include an Accept-Encoding header set to one or more of the following compression types:
    - gzip
    - deflate
    - bzip2
    - br

Note: This header allows the user agent to indicate which compression methods it supports to the origin server.

2. The web servers associated with your customer origin must support the compression method defined within the Accept-Encoding request header.

Basic Workflow

The process through which requested content is compressed is illustrated below.



HTTP Request/Response (Origin Server Compression)

Detailed Workflow

Detailed information on how requests are handled with regards to origin server compression is provided below.

1. **Requester:** A user submits a request for content. The manner in which this request is handled by our edge servers depends on whether the request includes the Accept-Encoding header.

Accept-Encoding Header	Description
Set to a supported compression method.	This type of request will be handled as described in step 2.
Set to an unsupported compression method.	This type of request is uncacheable. Our edge servers will always retrieve the requested content from your customer origin server.
Missing	This type of request will be served in an uncompressed format.

2. **POP:** An edge server on the POP closest to the client will check to see if the requested content has been cached and if it still has a valid TTL.
    - **Cache Miss:** If a cached version of the requested content is not found, then the request will be forwarded to an origin server. Proceed to step 3.
    - **Cache Hit & Matching Compression Method:** An edge server will immediately deliver the compressed content to the client.
    - **Cache Hit & Different Compression Method:** If the client requests a supported compression method that is different from the one used by the initial request, then an edge server will transcode the asset to the requested compression method.
    - **Uncompressed Cache Hit:** If the initial request caused the asset to be cached in an uncompressed format, then a check will be performed to see whether the request is eligible for edge server compression.
        - **Eligible:** The requested asset will be compressed as indicated in the **Edge Server Compression** section and then delivered to the client. Your caching policy dictates whether the compressed asset is eligible to be cached.
        - **Ineligible:** An edge server will immediately deliver the uncompressed content to the client.
3. **Origin Server:** The manner in which the origin server will handle the request depends on whether both the CDN and your web servers support the compression method specified in the Accept-Encoding header.

CDN	Customer Origin	Request Workflow
Supported compression method	Supported compression method	1. **Customer Origin:** It will compress the asset and send it to an edge server. 2. **Edge Server:** It will serve the compressed asset to the client. Your caching policy dictates whether the compressed asset is eligible to be cached.

CDN	Customer Origin	Request Workflow
Supported compression method	Unsupported compression method	1. **Customer Origin:** It will serve an uncompressed asset to an edge server. 2. **Edge Server:** It will: • Serve the uncompressed asset to the client. • Check to see whether the request is eligible for edge server compression. i. **Eligible:** The requested asset will be compressed as indicated in the **Edge Server Compression** section and then delivered to the client. Your caching policy dictates whether the compressed asset is eligible to be cached. ii. **Ineligible:** An edge server will immediately deliver the uncompressed content to the client. Your caching policy dictates whether the uncompressed asset is eligible to be cached.
Unsupported compression method	Supported compression method	1. **Customer Origin:** It will serve a compressed asset to an edge server. 2. **Edge Server:** It will serve the compressed asset to the client. However, it will not cache it.
Unsupported compression method	Unsupported compression method	1. **Customer Origin:** It will serve an uncompressed asset to an edge server. 2. **Edge Server:** It will serve the uncompressed asset to the client. However, it will not cache it.

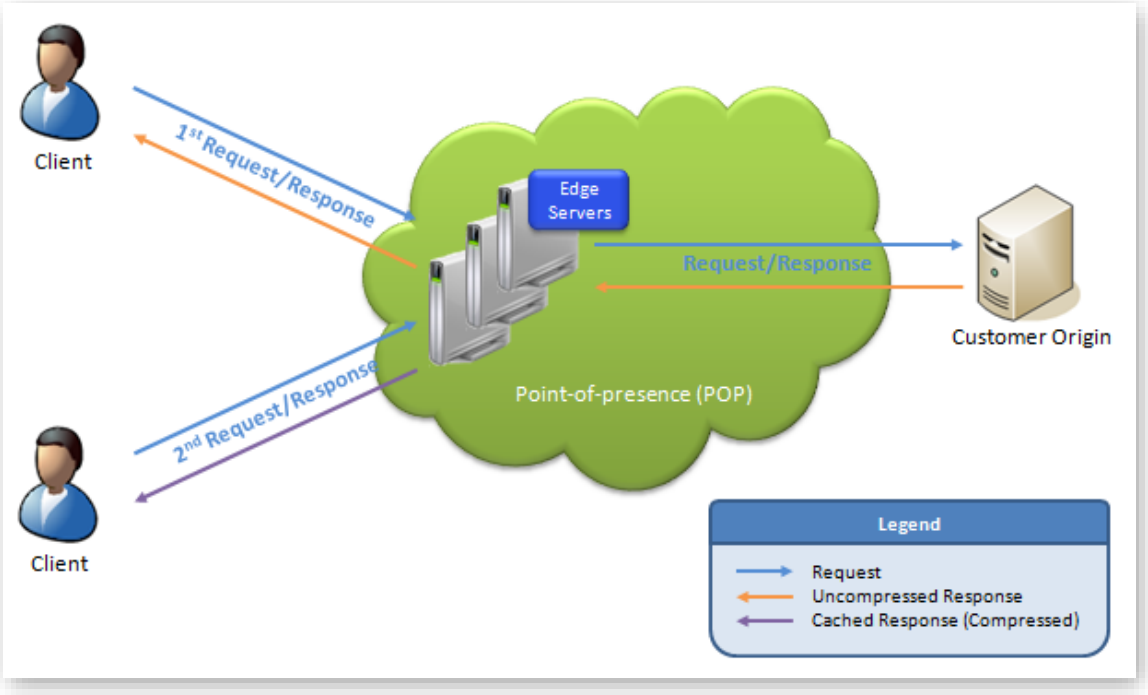
Edge Server Compression

Our edge servers can be configured to compress certain types of assets before they are delivered to your clients. The type of assets that will be compressed is determined by a list of content types that have been specified in the **File Types** option of the **Compression** page. Content type identifies an asset by type and subtype. For example, a content type of "text/plain" identifies text assets, while a content type of "text/html" identifies HTML assets.

Key information:

- Edge server compression requires the Accept-Encoding header and it must be set to a supported compression method. Supported compression methods are:
  - gzip
  - deflate
  - bzip2
- Content compressed by our edge servers will only be served from cache. As a result, edge server compression is only suitable for cacheable content.
- The **File Types** option accepts one or more content types that will be compressed.
- Multiple content types can be specified by separating each one with a comma.
- Compression can only be applied to content smaller than 3 MB. Larger assets will not be compressed by our servers.
- Certain types of content, such as images, video, and audio media assets (e.g., JPG, MP3, MP4, etc.), are already compressed. Additional compression on these types of assets may not significantly diminish file size. Therefore, it is recommended that you do not enable compression on these types of assets.
- Wildcard characters, such as asterisks, are not supported in the **File Types** option.

The process through which our edge servers compress content is illustrated below.



HTTP Requests/Responses (Edge Server Compression)

The process through which requested content is compressed is outlined below. This process assumes that a client performed a request that is eligible for edge server compression.

1. An edge server on the POP closest to the client will check to see if the requested content has been cached and if it still has a valid TTL.
    - **Compressed Cache Hit:** An edge server will immediately deliver the compressed content to the client.
    - **Uncompressed Cache Hit:** An edge server will compress the requested content, serve it to the client, and then cache the compressed asset.
    - **Cache Miss:** If the asset has not been cached, then the request will be forwarded to an origin server. Proceed to the next step.
2. The response from the origin server will be one of the following:
    - **Compressed Content:** An edge server will deliver the compressed content to the client. For more information, please refer to the **Origin Server Compression** section above.
    - **Uncompressed Content:** An edge server will serve the uncompressed content to the client. After which, the edge server will compress it and then cache the compressed version.

To set asset compression

1. Navigate to the **Compression** page. Load this page by finding the **HTTP Large** menu, pointing to **Cache Settings**, and then selecting **Compression**.
2. Make sure that the **Compression Enabled** option is selected.
3. In the **File Types** list, make sure that only the desired content types are listed. For each content type that is not currently listed, append a comma and the content type (i.e., type/subtype).
4. Click **Update** to save your changes.

Note: Keep in mind that it may take up to an hour for your changes to take effect.

Overwriting Cache Headers

Another way to override the default cache behavior is to overwrite the cache headers assigned to your assets. This can be accomplished through HTTP Rules Engine. This feature allows you to create custom rules that can be used to handle how our edge servers cache and grant access to assets requested through HTTP-based platforms. For detailed information on how to use HTTP Rules Engine, please refer to the **HTTP Rules Engine Administration Guide**, which is available to users that have purchased it from the MCC home page and the **Rules Engine** page.

Note: HTTP Rules Engine must be purchased separately. For more information, please contact your CDN account manager.

Bandwidth Throttling

Overview

Bandwidth throttling is the ability to limit the rate at which a user can download an asset. A common use case scenario for this capability occurs when you suspect that your customers are not watching your videos in their entirety. By reducing the speed at which data can be downloaded, you can ensure that you are only paying for the bandwidth traffic that is being used.

Configuring Bandwidth Throttling

Bandwidth throttling options are specified as query strings in a CDN or edge CNAME URL. The available query string parameters for configuring bandwidth throttling are **ec_rate** and **ec_prebuf**. Both of these parameters are explained below.

Parameter	Description
ec_rate	Determines the maximum download rate for a particular request in Kilobytes per second (KBps). If a customer's connection is faster than the specified value, then that customer's download rate will remain constant at the specified value. This parameter only accepts values equal to or higher than 32 KBps.
ec_prebuf	Determines the amount of data that can be downloaded before the ec_rate parameter takes effect. This creates a buffer before a user's download speed is throttled down. A buffer is useful to reduce the amount of stuttering or buffering issues that users with slow connections may experience as a result of bandwidth throttling. This buffer can be calculated by multiplying ec_prebuf, which should be specified in seconds, with the bit rate set in ec_rate. The resulting value will be the total amount of data that can be downloaded before bandwidth throttling takes effect.

Typically, bandwidth throttling options are passed as query string parameters. For example, the following URL would create a buffer of 640 KB and limit download speeds to 64 Kbps:

http://wpc.0001.edgecastcdn.net/000001/Videos/Demo.flv?ec_rate=64&ec_prebuf=10

Tip: If you take advantage of creating a buffer, then the recommended configuration is to set ec_rate to a value that matches the bit rate at which your video was encoded.

Note: Both of these parameters can be encrypted through Token-Based Authentication. This prevents a user from altering the values associated with them. For more information on how to encrypt bandwidth throttling parameters, please refer to the **Token-Based Authentication Administration Guide**.

HTTP Streaming Solutions

Overview

The HTTP Large platform supports various streaming solutions, such as:

- HTTP Progressive Download
- Dynamic Cloud Packaging

Please refer to the CDN Help Center to learn more about our streaming solutions.

Glossary

A

Asset

This term refers to a resource that contains header information and a body that can be served to clients. Examples of assets include files and dynamic content.

C

Cache

This term refers to the storage of data to improve data delivery performance. When used in reference to our CDN, it refers to the temporary storage of an asset on an edge server or an origin shield server. Cache increases the speed through which that particular edge server can deliver that asset for subsequent requests.

CDN

Our content delivery network (CDN) consists of points-of-presence (POPs) that are placed at critical network and geographical locations around the world. This allows us to place content at the edge of the Internet allowing for faster downloads by your end-users.

CDN Domain

This term refers to a domain name assigned to your account. In the following examples of CDN domains, *xxxx* represents your CDN account number.

- wac.*xxxx*.edgecastcdn.net
- wpc.*xxxx*.edgecastcdn.net
- fms.*xxxx*.edgecastcdn.net

CDN Origin

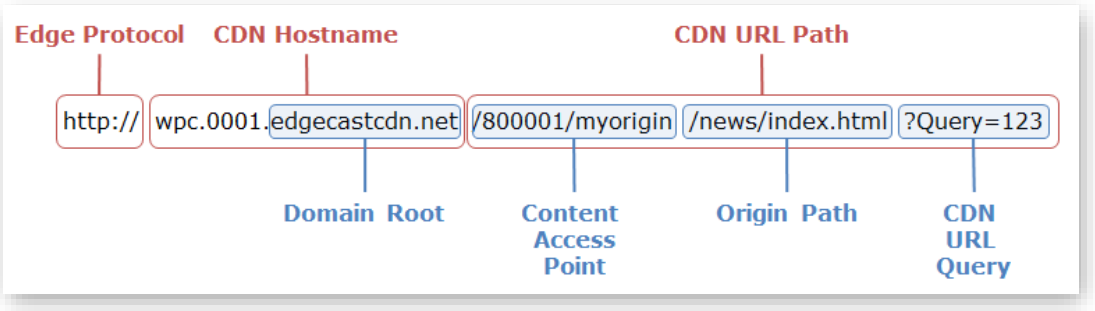
This term refers to a storage server on our CDN. Our CDN origin servers are in close proximity to our POPs, in order to provide optimal conditions for transferring data from a CDN origin server to your end-users via our POPs.

CDN Origin Identifier

This type of identifier in the CDN URL indicates that requested asset should be retrieved from the CDN origin server. A CDN origin identifier is indicated by "00" as the starting two numbers in the CDN URL path.

CDN URL

This type of URL identifies a location or an asset on our content delivery network. The following diagram indicates the different components in a CDN URL. Keep in mind that *xxxx* represents your CDN account number.



CDN URL Path

This term refers to the portion of the CDN URL that appears after the CDN domain. It provides the relative path to a folder or an asset on either a CDN or customer origin server. In the following examples of CDN URL paths, *xxxx* represents your CDN account number.

- /00*xxxx*
- /00*xxxx*/Videos/Sales/01/
- /00*xxxx*/Videos/Sales/01/Presentation01.flv

CDN/Edge CNAME URL Query

This term refers to the query string that appears after a question mark in a CDN or edge CNAME URL. If Token-Based Authentication is protecting the requested content, then a token value should appear directly after the question mark.

CNAME

A Canonical Name (CNAME) record is used to indicate that a domain name is an alias of another domain name. A CNAME record must be registered on a Domain Name System (DNS). This term should not be confused with edge CNAME.

Content Access Point

It provides a point of reference to any folder on a CDN or customer origin server. This relative path starts directly after the CDN domain. The proper syntax for a content access point is "/*yyxxxx*/*path*," where *yy* stands for the identifier and *xxxx* stands for the CDN account number. The term *path* is optional and stands for the path to the folder specified by an edge CNAME configuration.

Customer Origin

This term refers to a storage server that is external to our CDN. Assets can be delivered from your storage server to your end-users via our POPs.

Customer Origin Identifier

This type of identifier in the CDN URL indicates that requested asset should be retrieved from the customer origin server. A customer origin identifier is indicated by "80" as the starting two numbers in the CDN URL path.

D

Domain Root

This term identifies the top and second-level domains associated with the CDN domain name. An example of a domain root is "google.com."

Dynamic Streaming

This technology, also known as Adaptive Streaming, allows a player (e.g., Silverlight) to dynamically switch between bit rate streams of varying quality levels, in order to provide an optimal viewing experience based on a client's bandwidth and CPU usage.

E

Edge CNAME

This term refers to the mapping of a CNAME record to a directory on a CDN or customer origin server. The purpose of this mapping, which is only used by our CDN, is to establish a user-friendly alias for content served through the CDN. It relies upon your CNAME record being properly mapped on a DNS server.

Edge CNAME URL

This type of URL takes advantage of an edge CNAME to mask a CDN URL. This allows it to identify a location or an asset on our content delivery network using a more user-friendly URL. An edge CNAME URL is specific to the platform from which it was configured.

In the following examples, the domain assigned to the edge CNAME is "www.mydomain.com." In the first example, the edge CNAME references the following CDN URL: "http://wpc.*xxxx*.edgecastcdn.net/00*xxxx*." In the following two examples, the edge CNAME references the following CDN URL: "http://wpc.*xxxx*.edgecastcdn.net/00*xxxx*/Videos."

Edge CNAME URL	Points To
http://www.MyDomain.com/	http://wpc.*xxxx*.edgecastcdn.net/00*xxxx*/
http://www.MyDomain.com/Sales/01/	http://wpc.*xxxx*.edgecastcdn.net/00*xxxx*/Videos/Sales/01/
http://www.MyDomain.com/Sales/01/Presentation01.flv	http://wpc.*xxxx*.edgecastcdn.net/00*xxxx*/Videos/Sales/01/Presentation01.flv

Edge CNAME URL Path

This term refers to the portion of the edge CNAME URL that appears after the edge CNAME. It provides the relative path to a folder or an asset on a CDN or customer origin server. In the following examples of edge CNAME URL paths, the edge CNAME points to the following CDN URL: "http://wpc.*xxxx*.edgecastcdn.net/00*xxxx*/Videos."

Edge CNAME URL Path	Actual Edge CNAME URL
/Sales/01/	http://wpc.*xxxx*.edgecastcdn.net/00*xxxx*/Videos/Sales/01/
/Sales/01/Show01.flv	http://wpc.*xxxx*.edgecastcdn.net/00*xxxx*/Videos/Sales/01/Show01.flv

Edge Protocol

This term refers to the protocol (e.g., HTTP and RTMP) used in a CDN URL or an edge CNAME URL.

Edge Server

This type of server is located near the edge of the Internet where its close proximity to your end-users allows it to deliver data more quickly than normal Internet communications. Our edge servers are integral component of our POPs.

Encryption Key

Token-Based Authentication requires the use of an encryption key to encrypt and decrypt token values. There are two types of encryption keys, which are a primary and a backup key. Both of these keys can be used to encrypt and decrypt token values.

G

Global Key

This type of Live Authentication key can be used to authenticate all live streams. Only a single global key can be specified.

H

HTTP Large

This platform consists of dedicated edge servers that retrieve, cache, and serve large assets to your clients. These servers have been optimized to cache assets. A typical asset for the HTTP Large platform is larger than 300 KB.

HTTP Progressive Download

This method of streaming video content is performed through the HTTP protocol. Progressive downloads are not as secure as other streaming methods, since the entire asset will be stored on your client's computer. This allows your client to save and share your content with other users.

HTTP Small

This platform consists of dedicated edge servers that retrieve, cache, and serve smaller content to your clients. These servers have been optimized to index files. A typical asset for the HTTP Small platform is smaller than 300 KB.

I

Identifier

It identifies how a request will be routed through our CDN. Examples of identifiers are:

- **00:** CDN origin identifier
- **80:** Customer origin identifier

Ingest

This term refers to the process of capturing and transforming video into a stream.

Ingest Server

This term refers to the type of server that is dedicated to the process of capturing and transforming media into a stream. This type of server will then broadcast that stream throughout our CDN.

L

Live Authentication Key

This type of key authenticates a live stream before it is ingested by our publishing server. There are two types of Live Authentication keys, which are global and stream keys. A live authentication key must be specified when setting an encoder's stream setting. The proper notation is provided below.

```
StreamName?LiveAuthenticationKey
```

Live Ingestion Point

This term refers to the location on a server where our CDN can access encoded media. Our streaming services can ingest live streams via a publishing point.

Live StreamCast

This term refers to the streaming of a live media stream through our CDN. This type of streaming uses Real-Time Messaging Protocol (RTMP, RTMPE, RTMPT, or RTMPTE) to deliver video to your clients.

Load

This feature allows you to cache an asset on all of our POPs. This feature is unsupported for use with our live streaming solutions.

M

Media Control Center (MCC)

This web application is provided to help you manage all of your CDN needs. The major features that are available from the MCC are CDN configuration settings, cache management, file management, reports, and analytics. Additionally, the MCC allows you to configure your organization's settings, such as granting or denying access to the MCC. You can access the MCC through the following URL:

https://my.edgecast.com

O

Origin Path

It references a relative path to a folder or an asset in a CDN URL. This type of path follows the content access point.

Origin Server

This term refers to the servers that store the assets that will be distributed by our POPs. There are two types of origin servers, which are CDN origin and customer origin servers.

Origin Shield

This feature provides a layer of protection for your customer origin server by creating an intermediate caching layer between it and our edge servers. This caching layer resides on one or more of our point-of-presence (POPs). Requests that have not been previously cached on a POP will be channeled through the closest origin shield server. The origin shield server will then either serve a cached version of the requested content or retrieve it from your customer origin server. This feature reduces the amount of bandwidth used on your customer origin server, since most requests will be handled by the origin shield server.

P

Player URL

A media player uses this type of URL to stream content. It identifies the location of the streamer on the live ingestion point.

Point-of-Presence (POP)

A point-of-presence, or data center, is an access point to the Internet. The main components of a POP are edge servers, CDN origin servers, and publishing servers.

Pre-Cached

A pre-cached asset means that it has been loaded to all of our POPs. Pre-caching your assets allows even quicker content delivery to your clients, since it ensures that the requested asset will not have to be retrieved from the origin server.

Publishing Point

This term refers to the location on the publishing server to which your encoder will broadcast encoded media.

Publishing Server

This term refers to a CDN server that will redistribute encoded media as a streamer that will be broadcast to your end-users via our POPs.

Purge

This feature allows you to remove the cached version of an asset from all of our edge servers and origin shield servers. A purge can be performed on a folder or an individual asset.

Push Stream

This type of stream requires your encoder to send, or push, encoded video to a CDN server. From there, our server will create a stream and deliver it to clients that request it.

Q

Query String

Additional data can be appended to a URL (e.g., http://www.server.com/index.html?Data=xyz). This information can be used in a variety of ways. Our CDN allows you to leverage this information to determine how content will be cached. Additionally, you can choose to store query string information in our log files.

R

Request

A request consists of a set of headers and a body sent from a client. This header data and the body define the requested content. Typically, a request is sent from a client to an edge server. If the requested content is not found, then our edge servers will forward this request to an origin server.

Response

A response consists of the headers and the body sent from a server responding to a request. If an origin server is returning a response, then this response will be sent to an edge server. The edge server will then forward the response to a client.

S

Stream

A stream consists of the delivery of audio/video content in a format that allows your clients to play it back through a multimedia player.

Stream Key

This type of Live Authentication key can only authenticate a live stream when it is published to the path associated with it.

T

Time to Live (TTL)

This term refers to the amount of time that a cached asset is still considered fresh. Our edge servers will continue to serve a cached version of an asset while its TTL has not expired. An asset's TTL is calculated by the Cache-Control and Expires headers associated with the response sent by a CDN or customer origin server.

Token

A token value must be provided when Token-Based Authentication has been applied to the requested content. Each token value contains a set of encoded requirements that must be met before content delivery may take place. A token value may be specified as a query string in the request URL (e.g., sales.pdf?1234567890AB).

Token-Based Authentication

This capability provides the means for defining the set of content that will require authentication prior to delivery. Authentication takes place via an encoded token value that must be included in the request URL. This token value is then decrypted on an edge server. The requested content will only be delivered when the user meets the requirement(s) defined in the token.