Amazon CloudFront

Developer Guide

API Version 2014-05-31

Amazon CloudFront: Developer Guide

Table of Contents

What Is Amazon CloudFront? 1
- How CloudFront Delivers Content 4
- Locations and IP Address Ranges of CloudFront Edge Servers 6

CloudFront Billing and Usage Reports 7
- AWS Billing Report for CloudFront 9
- CloudFront Usage Report 10
- CloudFront Usage Charts 11
  - How the Usage Charts Are Related to Data in the CloudFront Usage Report 12
- Interpreting Your AWS Bill and the CloudFront Usage Report 14

Getting Started 17
- Step 1: Sign up for Amazon Web Services 17
- Step 2: Upload your content to Amazon S3 and grant object permissions 17
- Step 3: Create a CloudFront Web Distribution 19
- Step 4: Test your links 24

Working with Distributions 25
- Overview of Web and RTMP Distributions 26
  - Web Distributions 26
  - RTMP Distributions 27
- Creating Web and RTMP Distributions 27
- Listing, Viewing, and Updating CloudFront Distributions 27
- Deleting a Distribution 28
- Using Alternate Domain Names (CNAMEs) 29
  - Using the * Wildcard in Alternate Domain Names 29
  - Restrictions on Using Alternate Domain Names 30
  - Adding an Alternate Domain Name 30
- Choosing the Price Class for a CloudFront Distribution 32
- Using CloudFront with Amazon S3 33
- Changes to the CloudFront API 34

Working with Web Distributions 36
- Task List for Creating a Web Distribution 36
- Creating a Web Distribution Using the CloudFront Console 37
- Testing Your Web Distribution 38
- Using Amazon S3 Origins and Custom Origins for Web Distributions 39
  - Using Amazon S3 Buckets for Your Origin 39
  - Using Amazon EC2 or Other Custom Origins 40
- Values that You Specify When You Create or Update a Web Distribution 40
  - Delivery Method 42
  - Origin Settings 42
  - Cache Behavior Settings 44
  - Distribution Details 50
  - Custom Error Pages and Error Caching 53
  - Restrictions 54
- Values that CloudFront Displays in the Console When You Create or Update a Web Distribution 55
  - Distribution ID (General Tab) 55
  - Distribution Status (General Tab) 55
  - Last Modified (General Tab) 55
  - Domain Name (General Tab) 55
- Requirements and Recommendations for Using Amazon EC2 and Other Custom Origins 56
- Restricting the Geographic Distribution of Your Content 56
- Configuring On-Demand Smooth Streaming 57
- Configuring On-Demand Progressive Downloads 58
- Configuring On-Demand Apple HTTP Live Streaming (HLS) 59

Working with RTMP Distributions 60
- How RTMP Distributions Work 60
- Task List for Streaming Media Files Using RTMP 62

Creating an RTMP Distribution Using the CloudFront Console ... 63
Values that You Specify When You Create or Update an RTMP Distribution ... 63
Origin Domain Name (Amazon S3 Bucket) ... 64
Restrict Bucket Access (Amazon S3 Only) ... 65
Origin Access Identity (Amazon S3 Only) ... 65
Comment for New Identity(Amazon S3 Only) ... 65
Your Identities (Amazon S3 Only) ... 65
Grant Read Permissions on Bucket (Amazon S3 Only) ... 65
Price Class ... 65
Alternate Domain Names (CNAMEs) ... 66
Logging ... 66
Bucket for Logs ... 66
Log Prefix ... 66
Comment ... 66
Distribution State ... 66
Restrict Viewer Access (Use Signed URLs) ... 67
Trusted Signers ... 67
AWS Account Numbers ... 67
Values that CloudFront Displays in the Console When You Create or Update an RTMP Distribution ... 68
Distribution ID ... 68
Status ... 68
Last Modified ... 68
Domain Name ... 68
Configuring the Media Player ... 69
MPEG Files ... 69
Using an Amazon S3 Bucket as the Origin for an RTMP Distribution ... 69
Creating Multiple RTMP Distributions for an Origin Server ... 70
Restricting Access Using Crossdomain.xml ... 70
Error Codes for RTMP Distributions ... 71
Troubleshooting RTMP Distributions ... 71
Working with Objects ... 72
Format of URLs for CloudFront Objects ... 72
Format of Public URLs for Objects in Amazon S3 ... 73
Format of Public URLs for Objects in a Custom Origin ... 73
How Public URLs Affect the Invalidation of Directories ... 74
Format of Signed URLs ... 74
How CloudFront Processes HTTP and HTTPS Requests ... 74
Configuring CloudFront to Cache Based on Query String Parameters ... 75
Query String Parameters and Web Distributions ... 75
Query String Parameters and RTMP Distributions ... 76
Configuring CloudFront to Cache Objects Based on Cookies ... 76
Cookies and Web Distributions ... 77
Cookies and RTMP Distributions ... 78
Configuring CloudFront to Cache Objects Based on Request Headers ... 78
Headers and Web Distributions ... 78
Headers and RTMP Distributions ... 81
Adding, Removing, or Replacing Objects in a Distribution ... 81
Adding Objects that You Want CloudFront to Distribute ... 82
Updating Existing Objects Using Versioned Object Names ... 82
Updating Existing Objects Using the Same Object Names ... 83
Specifying How Long Objects Stay in a CloudFront Edge Cache (Expiration) ... 83
Invalidating Objects (Web Distributions Only) ... 87
Customizing Error Responses ... 92
Creating or Updating a Cache Behavior for Custom Error Pages ... 93
Changing Response Codes ... 93
Controlling How Long CloudFront Caches Errors ... 94
How CloudFront Responds When a Custom Error Page Is Unavailable ... 94

Pricing for Custom Error Pages 94
Configuring Error Response Behavior 95
How CloudFront Processes Partial Requests for an Object (Range GETs) 96
Specifying a Default Root Object (Web Distributions Only) 96
Serving Compressed Files 98
How CloudFront Serves Compressed Content from a Custom Origin 99
Serving Compressed Files from Amazon S3 99
Choosing the File Types to Compress 100
Request and Response Behavior 101
Request and Response Behavior for Amazon S3 Origins 101
How CloudFront Processes and Forwards Requests to Your Amazon S3 Origin Server 101
How CloudFront Processes Responses from Your Amazon S3 Origin Server 104
Request and Response Behavior for Custom Origins 106
How CloudFront Processes and Forwards Requests to Your Custom Origin Server 106
How CloudFront Processes Responses from Your Custom Origin Server 113
How CloudFront Processes and Caches HTTP 4xx and 5xx Status Codes 115
How CloudFront Processes Errors When You Have Configured Custom Error Pages 116
How CloudFront Processes Errors When You Have Not Configured Custom Error Pages 116
HTTP 4xx and 5xx Status Codes that CloudFront Caches 117
Serving Private Content through CloudFront 118
Overview of Private Content 118
How Private Content Works 119
Using an HTTP Server for Private Content 120
Choosing How Long Signed URLs Are Valid 120
Sample Code and Third-Party Tools 120
Task List: Serving Private Content 122
Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content 123
Creating a CloudFront Origin Access Identity and Adding it to Your Distribution 123
Granting the Origin Access Identity Permission to Read Objects in Your Amazon S3 Bucket 125
Specifying the AWS Accounts That Can Create Signed URLs (Trusted Signers) 128
Creating CloudFront Key Pairs for Your Trusted Signers 129
Reformatting the CloudFront Private Key (.NET and Java Only) 129
Adding Trusted Signers to Your Distribution 130
Verifying that Trusted Signers Are Active (Optional) 132
Rotating CloudFront Key Pairs 132
Overview of Signed URLs 134
Signed URLs that You Create Using a Canned Policy 134
Signed URLs that You Create Using a Custom Policy 134
The Parts of a Signed URL 135
When Does CloudFront Check the Expiration Date and Time in a Signed URL? 137
Creating a Signed URL Using a Canned Policy 138
Creating a Signature for a Canned Policy 139
Creating a Signed URL Using a Custom Policy 143
Creating a Policy Statement for a Custom Policy 144
Example Policy Statements for a Custom Policy 146
Creating a Signature for a Custom Policy 148
Using Linux Commands and OpenSSL for Base64-Encoding and Encryption 151
Base64-Encoding the Policy Statement 151
Converting the Policy Statement into a Signature 151
Code and Examples for Creating a Signature for a Signed URL 153
Create a URL Signature Using Perl 153
Create a URL Signature Using PHP 154
Create a URL Signature Using C# and the .NET Framework 157
Create a URL Signature Using Java 165
Using an HTTPS Connection to Access Your Objects 168
How CloudFront Works with HTTPS Connections 169
How to Require HTTPS for Communication between Viewers, CloudFront, and Your Origin 169

Using Alternate Domain Names and HTTPS ... 171
Choosing How CloudFront Serves HTTPS Requests ... 171
Requirements and Limits on Using SSL Certificates with CloudFront ... 172
To use alternate domain names with HTTPS ... 174
Determining the Size of the Public Key in an SSL Certificate ... 175
Rotating SSL Certificates ... 175
Reverting from a Custom SSL Certificate to the Default CloudFront Certificate ... 176
Switching from a Custom SSL Certificate with Dedicated IP Addresses to SNI ... 177
Charges for HTTPS Connections ... 177
Using IAM to Control Access to CloudFront Resources ... 178
CloudFront Resources ... 178
CloudFront Actions ... 179
Web Distributions ... 179
RTMP Distributions ... 179
Invalidations ... 179
Origin Access Identities ... 180
Policy Keys ... 180
Example Policies for CloudFront ... 180
Access Logs ... 182
Overview ... 182
Analyzing Access Logs ... 184
Bucket and File Ownership ... 184
How to Change Logging Settings ... 184
How to Delete Log Files from an Amazon S3 Bucket ... 185
File Name Format and Timing of File Delivery ... 185
Log File Format ... 185
Web Distribution Log File Format ... 186
RTMP Distribution Log File Format ... 189
Charges for Access Logs ... 191
Capturing API Requests with CloudTrail ... 192
CloudFront Information in CloudTrail Log Files ... 192
Understanding CloudFront Log File Entries ... 193
Troubleshooting ... 198
I can't view the files in my web distribution. ... 198
Did you sign up for both CloudFront and Amazon S3? ... 198
Are your Amazon S3 bucket and object permissions set correctly? ... 198
Is your alternate domain name (CNAME) correctly configured? ... 199
Are you referencing the correct URL for your CloudFront distribution? ... 199
Do you need help troubleshooting a custom origin? ... 199
I can't view the files in my RTMP distribution. ... 200
Error Message: Certificate: <certificate-id> is being used by CloudFront. ... 200
Load Testing CloudFront ... 201
CloudFront Tutorials ... 202
Live Streaming ... 202
Geoblocking ... 202
RTMP Streaming ... 202
Live HTTP Streaming Using CloudFront and Adobe Media Server 5.0 ... 202
Overview ... 203
Steps to Configure Live Streaming ... 204
Creating an Amazon Web Services Account ... 204
Creating an Amazon EC2 Key Pair ... 204
Subscribing to Adobe Media Server ... 205
Creating an AWS CloudFormation Stack for Live Streaming ... 206
Verifying that Adobe Media Server Is Running ... 207
Setting Up Adobe Flash Media Live Encoder to Publish a Live Stream ... 208
Embedding Strobe Media Playback for an Amazon CloudFront Live HTTP Stream in a Web Application ... 211
Deleting an AWS CloudFormation Stack and an Amazon EBS Volume for Live Streaming ... 212

Frequently Asked Questions 213
Additional Documentation 219
Live Smooth Streaming Using Amazon CloudFront and IIS Media Services 4.1 220
Overview of Live Smooth Streaming with Amazon Web Services 220
Creating an Amazon Web Services Account 221
Creating an Amazon EC2 Key Pair 221
Creating an AWS CloudFormation Stack for Live Smooth Streaming 222
Verifying that Your Amazon EC2 Windows Server Instance Is Running 225
Getting Your Windows Password 225
Encoding Your Live Stream 226
Viewing Your Live Smooth Stream 231
Deleting Your AWS CloudFormation Live Smooth Streaming Stack 231
Frequently Asked Questions 232
Additional Documentation 234
Live Streaming with Wowza Streaming Engine 4.0 235
Creating an Amazon Web Services Account 235
Creating an Amazon EC2 Key Pair 236
Getting a License for Wowza Streaming Engine 4.0 236
Subscribing to Wowza Streaming Engine 4.0 through AWS Marketplace 236
Creating an AWS CloudFormation Stack for Live Streaming 237
Verifying that Wowza Streaming Engine 4.0 Is Running 238
Setting Up an Encoder to Publish a Live Stream 239
Playing the Live Stream in a Web Application 240
Deleting an AWS CloudFormation Stack for Live Streaming 241
Frequently Asked Questions 241
Additional Documentation 243
Restricting Access to Files in a CloudFront Distribution Based on Geographic Location (Geoblocking) 244
Creating an Amazon Web Services Account 246
Sample Code for Digital Element 246
Sample Code for MaxMind 256
Frequently Asked Questions 266
Additional Services and Documentation 267
On-Demand Video Streaming Using CloudFront and Adobe Flash Player 269
Creating an Amazon S3 Bucket 269
Creating CloudFront Web and RTMP Distributions 269
Creating a Flash Project Using Adobe Flash Builder 270
Uploading Media and Flash Builder Files to an Amazon S3 Bucket 272
Playing the Media File 273
On-Demand Video Streaming Using CloudFront and Flowplayer for Adobe Flash 274
Uploading Media and Flowplayer Files to an Amazon S3 Bucket 274
Creating CloudFront Web and RTMP Distributions 275
Embedding Video in an HTML Page 276
On-Demand Video Streaming Using CloudFront and JW Player 279
Uploading Media and JW Player Files to an Amazon S3 Bucket 279
Creating CloudFront Web and RTMP Distributions 280
Embedding Video in a Web Page 281
Uploading the HTML File and Playing the Video 282
Resources 283
Additional Amazon CloudFront Documentation 283
Getting Support 284
CloudFront Developer Tools and SDKs 284
Using CloudFront Logging 284
Additional Tips from the Amazon Web Services Blog 284
Invalidating Objects 285
Distributing Streaming Media 285
Tools for Configuring Private Content 285
Using CloudFront with a Content Management System 286

Document History 287
AWS Glossary 295

What Is Amazon CloudFront?

Topics

- How CloudFront Delivers Content (p. 4)
- Locations and IP Address Ranges of CloudFront Edge Servers (p. 6)

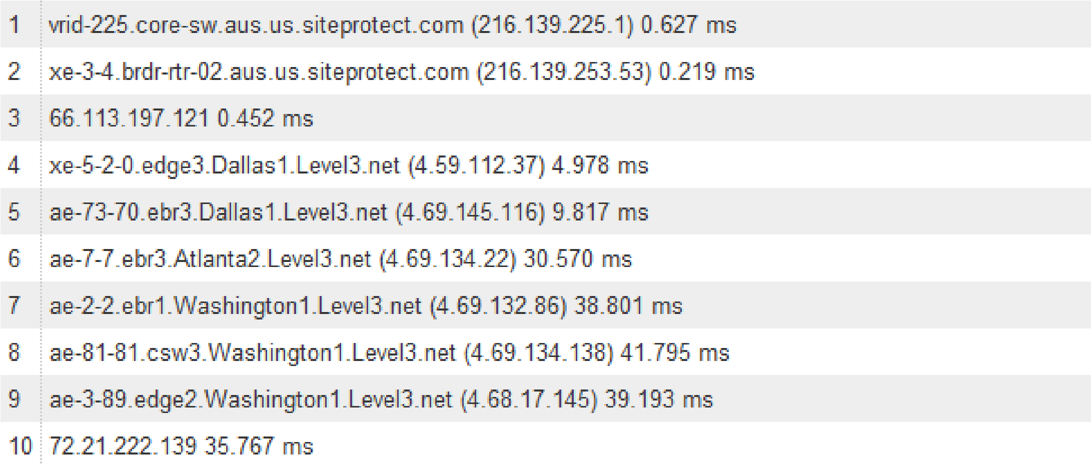
CloudFront is a web service that speeds up distribution of your static and dynamic web content, for example, .html, .css, .php, and image files, to end users. CloudFront delivers your content through a worldwide network of data centers called edge locations. When a user requests content that you're serving with CloudFront, the user is routed to the edge location that provides the lowest latency (time delay), so content is delivered with the best possible performance. If the content is already in edge location with the lowest latency, CloudFront delivers it immediately. If the content is not currently in that edge location, CloudFront retrieves it from an Amazon S3 bucket or an HTTP server (for example, a web server) that you have identified as the source for the definitive version of your content.

This concept is best illustrated by an example. Suppose you're serving the following image from a traditional web server, not from CloudFront:

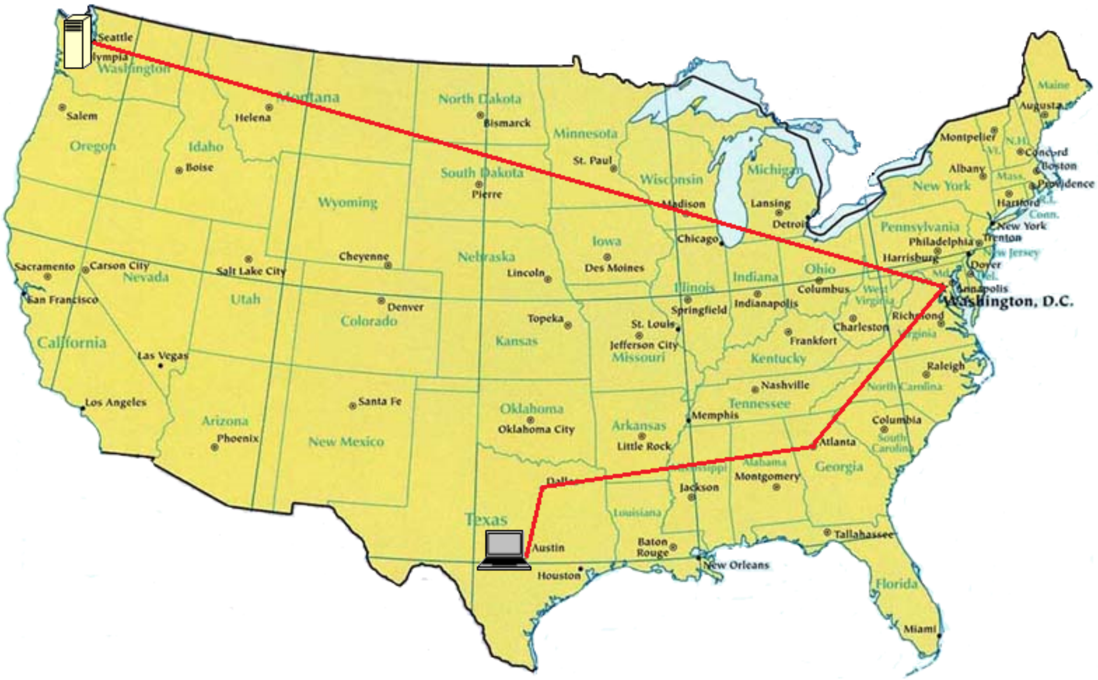
(The image is owned by NASA and comes from the Visible Earth website, http://visibleearth.nasa.gov/.)

You're serving the image using the URL `http://example.com/globe_west_540.png`. Your users can easily navigate to this URL and see the image, but they probably don't know that their request was routed from one network to another—through the complex collection of interconnected networks that comprise the Internet—until the image was found.

Further suppose that the web server from which you're serving the image is in Seattle, Washington, USA, and that a user in Austin, Texas, USA requests the image. The traceroute list below (courtesy of www.WatchMouse.com) shows one way that this request could be routed.



```
1  vrid-225.core-sw.aus.us.siteprotect.com (216.139.225.1) 0.627 ms
2  xe-3-4.brdr-rtr-02.aus.us.siteprotect.com (216.139.253.53) 0.219 ms
3  66.113.197.121 0.452 ms
4  xe-5-2-0.edge3.Dallas1.Level3.net (4.59.112.37) 4.978 ms
5  ae-73-70.ebr3.Dallas1.Level3.net (4.69.145.116) 9.817 ms
6  ae-7-7.ebr3.Atlanta2.Level3.net (4.69.134.22) 30.570 ms
7  ae-2-2.ebr1.Washington1.Level3.net (4.69.132.86) 38.801 ms
8  ae-81-81.csw3.Washington1.Level3.net (4.69.134.138) 41.795 ms
9  ae-3-89.edge2.Washington1.Level3.net (4.68.17.145) 39.193 ms
10 72.21.222.139 35.767 ms
```



Map courtesy of the University of Texas Libraries, The University of Texas at Austin

In this example, the request was routed 10 times within the United States before the image was retrieved, which is not an unusually large number of hops. If your user were in Europe, the request would be routed through even more networks to reach your server in Seattle. The number of networks and the distance that the request and the image must travel have a significant impact on the performance, reliability, and availability of the image.

CloudFront speeds up the distribution of your content by routing each user request to the edge location that can best serve your content. Typically, this is the CloudFront edge location that provides the lowest latency. This dramatically reduces the number of networks that your users' requests must pass through, which improves performance. Users get lower latency—the time it takes to load the first byte of the object—and higher data transfer rates. You also get increased reliability and availability because copies of your objects are now held in multiple edge locations around the world.

For a list of the locations of CloudFront edge servers, see The Amazon CloudFront Global Edge Network on the CloudFront Product Details page.

How CloudFront Delivers Content

After some initial setup, CloudFront works invisibly to speed up delivery of your content. This overview includes both the steps you perform before your first user accesses your application or website and how CloudFront serves your content when configuration is complete.

Setting up CloudFront involves a few simple steps:

How You Configure CloudFront to Deliver Your Content

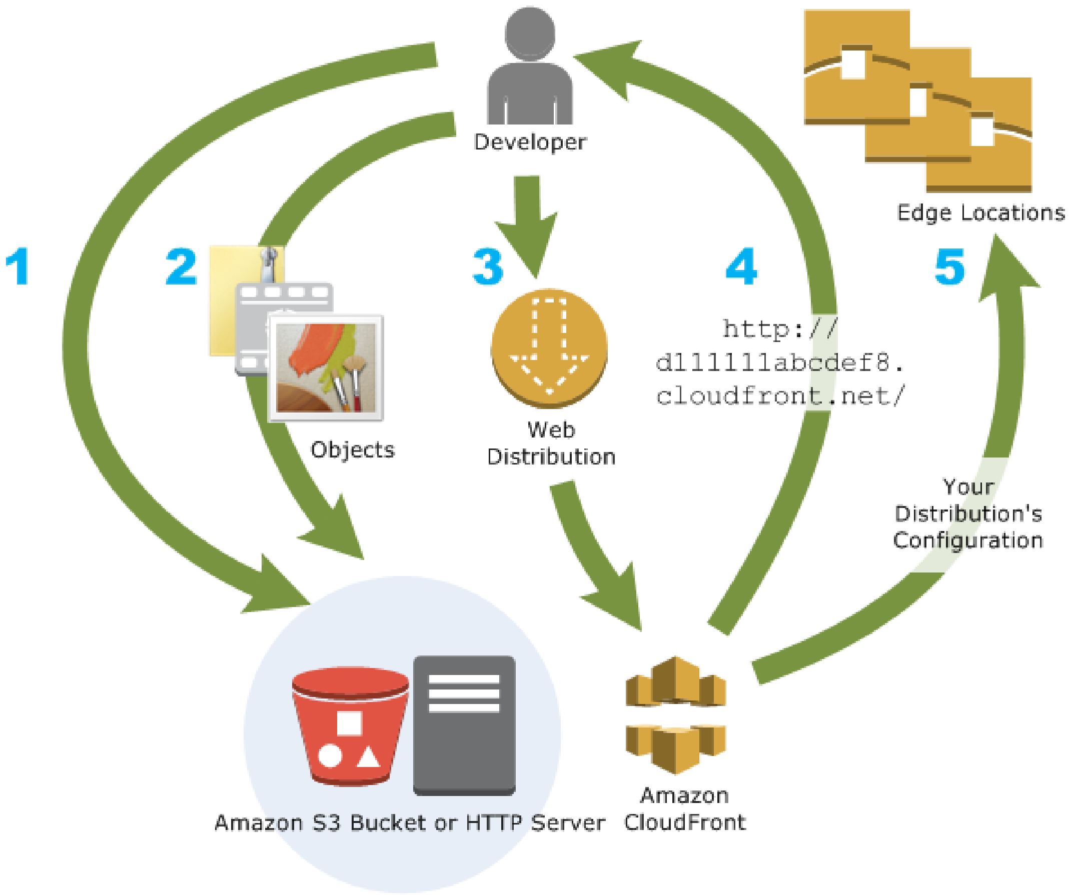
1. You configure your **origin servers**, from which CloudFront gets your files for distribution from CloudFront edge locations all over the world.

   An origin server stores the original, definitive version of your objects. If you're serving content over HTTP, your origin server is either an Amazon S3 bucket or an HTTP server, such as a web server. Your HTTP server can be running on an Amazon Elastic Compute Cloud (Amazon EC2) instance or on a server that you manage; these servers are also known as custom origins.

   If you're distributing media files on demand using the Adobe Media Server RTMP protocol, your origin server is always an Amazon S3 bucket.
2. You upload your files to your origin servers. Your files, also known as **objects**, typically include web pages, images, and media files, but can be anything that can be served over HTTP or a supported version of Adobe RTMP, the protocol used by Adobe Flash Media Server.

   If you're using an Amazon S3 bucket as an origin server, you can make the objects in your bucket publicly readable, so anyone who knows the CloudFront URLs for your objects can access them. You also have the option of keeping objects private and controlling who accesses them. See Serving Private Content through CloudFront (p. 118).
3. You create a CloudFront **distribution**, which tells CloudFront which origin servers to get your files from when users request the files through your web site or application. At the same time, you specify details such as whether you want CloudFront to log all requests and whether you want the distribution to be enabled as soon as it's created.
4. CloudFront sends your distribution's configuration (but not your content) to all of its **edge locations**—collections of servers in geographically dispersed data centers where CloudFront caches copies of your objects.
5. As you develop your website or application, you use the domain name that CloudFront provides for your URLs. For example, if CloudFront returns `d111111abcdef8.cloudfront.net` as the domain name for your distribution, the URL for logo.jpg in your Amazon S3 bucket (or in the root directory on an HTTP server) will be `http://d111111abcdef8.cloudfront.net/logo.jpg`.

   You can also configure your CloudFront distribution so you can use your own domain name. In that case, the URL might be `http://www.example.com/logo.jpg`.
6. Optionally, you can configure your origin server to add headers to the files; the headers indicate how long you want the files to stay in the cache in CloudFront edge locations. By default, each object stays in an edge location for 24 hours before it expires. The minimum **expiration time** is 0 seconds; there isn't a maximum expiration time limit. For more information, see Specifying How Long Objects Stay in a CloudFront Edge Cache (Expiration) (p. 83).


Developer
Edge Locations
1
2
3
4
5
Objects
Web Distribution
http://d111111abcdef8.cloudfront.net/
Your Distribution's Configuration
Amazon S3 Bucket or HTTP Server
Amazon CloudFront

How CloudFront Delivers Content to Your Users

Once you configure CloudFront to deliver your content, here's what happens when users request your objects:

1. A user accesses your website or application and requests one or more objects, such as an image file and an HTML file.
2. DNS routes the request to the CloudFront edge location that can best serve the user's request, typically the nearest CloudFront edge location in terms of latency, and routes the request to that edge location.
3. In the edge location, CloudFront checks its cache for the requested files. If the files are in the cache, CloudFront returns them to the user. If the files are *not* in the cache, it does the following:

    a. CloudFront compares the request with the specifications in your distribution and forwards the request for the files to the applicable origin server for the corresponding file type—for example, to your Amazon S3 bucket for image files and to your HTTP server for the HTML files.

    b. The origin servers send the files back to the CloudFront edge location.

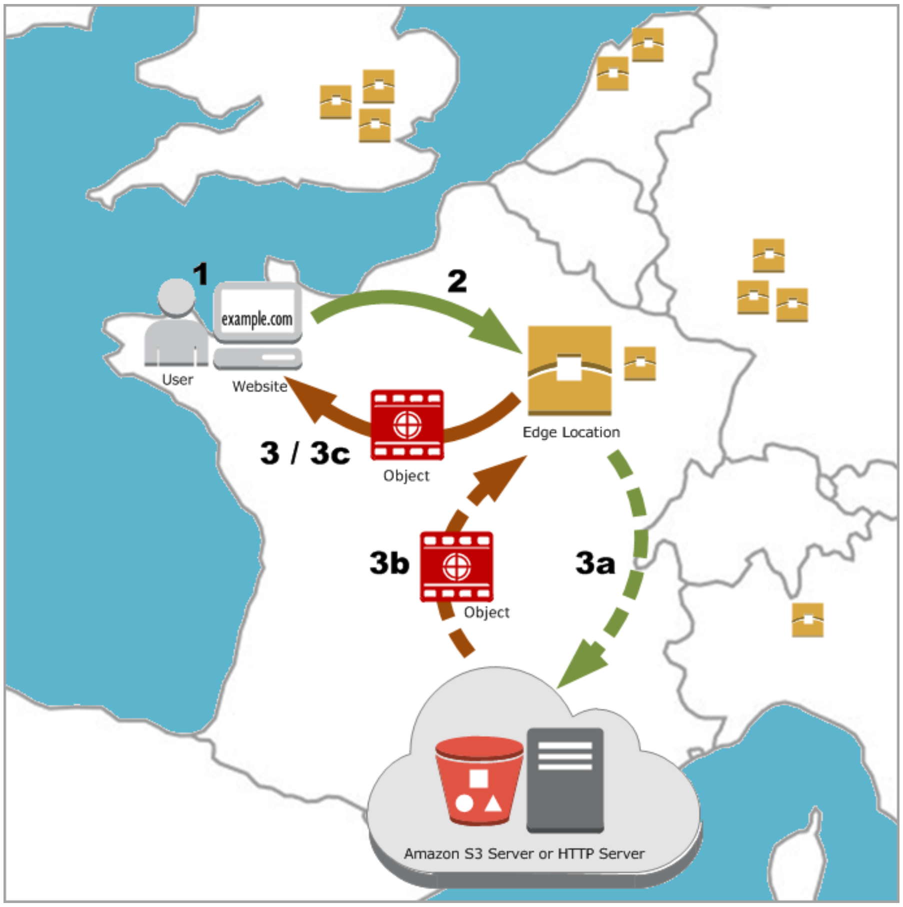
    c. As soon as the first byte arrives from the origin, CloudFront begins to forward the files to the user. CloudFront also adds the files to the cache in the edge location for the next time someone requests those files.

4. After an object has been in an edge cache for 24 hours or for the duration specified in your file headers, CloudFront does the following:

a. CloudFront forwards the next request for the object to your origin to determine whether the edge location has the latest version.

b. If the version in the edge location is the latest, CloudFront delivers it to your user.

   If the version in the edge location is not the latest, your origin sends the latest version to CloudFront, and CloudFront delivers the object to your user and stores the latest version in the cache at that edge location.



Locations and IP Address Ranges of CloudFront Edge Servers

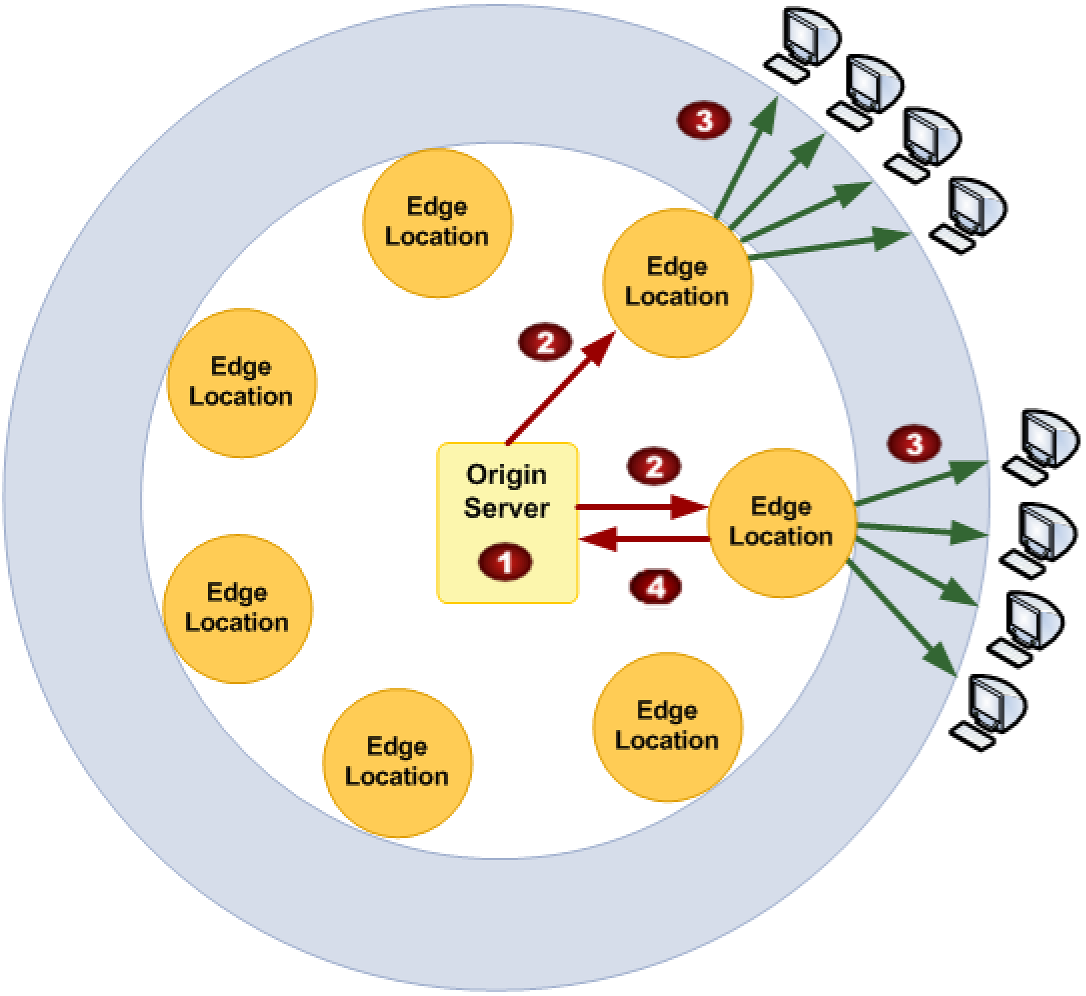
For a list of the locations of CloudFront edge servers, see The Amazon CloudFront Edge Network on the Amazon CloudFront detail page.

For a list of the IP address ranges of CloudFront edge servers, see Amazon CloudFront Public IP Ranges on the Amazon CloudFront discussion forum.

CloudFront Billing and Usage Reports

Amazon CloudFront is designed so you don't have to pay any up-front fees or commit to how much content you'll have. As with the other AWS services, you pay as you go and pay only for what you use.

The following diagram and table summarize the charges to use CloudFront.



Your monthly bill from AWS separates your usage and dollar amounts by AWS service and function. As a result, you see some charges for storing objects with Amazon S3 (if you are using Amazon S3 as your origin server) (1), some charges for data transfer between your bucket and your edge location (2), and some charges for serving data from CloudFront (3).

	Charge	Comments
1	Storage in an Amazon S3 origin server	You pay normal Amazon S3 storage charges to store objects in your bucket; the charges appear in the Amazon S3 portion of your AWS statement.
2	Copying objects to edge locations	If using an Amazon S3 origin server, you incur the normal Amazon S3 charges for GET requests and for data transfer out. CloudFront copies an object to an edge location only if there is demand for that object at that edge location. The data transfer charges appear in the AWS Data Transfer portion of your AWS statement.

	Charge	Comments
3	Serving objects from edge locations	You incur CloudFront charges for requests and data transfer out, which are lower than the corresponding Amazon S3 charges. The CloudFront charges appear in the CloudFront portion of your AWS statement. For more information, see Amazon CloudFront Pricing.
4	Submitting data to your origin	You incur CloudFront charges when users transfer data to your origin, which includes `DELETE`, `OPTIONS`, `PATCH`, `POST`, and `PUT` requests. The CloudFront charges appear in the CloudFront portion of your AWS statement. For more information, see Amazon CloudFront Pricing.

> **Note**
> You also incur a surcharge for HTTPS requests. For more information, see Amazon CloudFront Pricing.

AWS provides two usage reports for CloudFront:

- The billing report is a high-level view of all of the activity for the AWS services that you're using, including CloudFront. For more information, see AWS Billing Report for CloudFront (p. 9).
- The usage report is a summary of activity for a specific service, aggregated by hour, day, or month. For more information, see CloudFront Usage Report (p. 10).

In addition, you can view usage charts that provide a graphical representation of your CloudFront usage. For more information, see CloudFront Usage Charts (p. 11).

AWS Billing Report for CloudFront

You can view a summary of your AWS usage and charges, listed by service, on the Bills page in the AWS Management Console.

You can also download a more detailed version of the report in CSV format. The detailed billing report includes the following values that are applicable to CloudFront:

- **ProductCode** — `AmazonCloudFront`
- **UsageType** — One of the following values
  - A code that identifies the type of data transfer
  - `Invalidations`
  - `SSL-Cert-Custom`

  For more information, see Interpreting Your AWS Bill and the CloudFront Usage Report (p. 14).
- **ItemDescription** — A description of the billing rate for the **UsageType**.
- **Usage Start Date/Usage End Date** — The day that the usage applies to, in Coordinated Universal Time (UTC).
- **Usage Quantity** — One of the following values:
  - The number of requests during the specified time period
  - The amount of data transferred in gigabytes
  - The number of objects invalidated
  - The sum of the prorated months that you had SSL certificates associated with enabled CloudFront distributions. For example, if you have one certificate associated with an enabled distribution for an

entire month and another certificate associated with an enabled distribution for half of the month, this value will be 1.5.

To display summary billing information and download the detailed billing report

1. Sign in to the AWS Management Console at https://console.aws.amazon.com/console/home.
2. In the title bar, click your IAM user name, and click **Billing & Cost Management**.
3. In the navigation pane, click **Bills**.
4. To view summary information for CloudFront, under **Details**, click **CloudFront**.
5. To download a detailed billing report in CSV format, click **Download CSV**, and follow the on-screen prompts to save the report.

CloudFront Usage Report

AWS provides a CloudFront usage report that is more detailed than the billing report but less detailed than CloudFront access logs. The usage report provides aggregate usage data by hour, day, or month; and it lists operations by region and usage type, such as data transferred out of the Australia region.

The CloudFront usage report includes the following values:

- **Service** — `AmazonCloudFront`
- **Operation** — HTTP method. Values include `DELETE`, `GET`, `HEAD`, `OPTIONS`, `PATCH`, `POST`, and `PUT`.
- **UsageType** — One of the following values
  - A code that identifies the type of data transfer
  - `Invalidations`
  - `SSL-Cert-Custom`

  For more information, see Interpreting Your AWS Bill and the CloudFront Usage Report (p. 14).
- **Resource** — Either the ID of the CloudFront distribution associated with the usage or the certificate ID of an SSL certificate that you have associated with a CloudFront distribution.
- **StartTime/EndTime** — The day that the usage applies to, in Coordinated Universal Time (UTC).
- **UsageValue** — (1) The number of requests during the specified time period or (2) the amount of data transferred in bytes.

If you're using Amazon S3 as the origin for CloudFront, consider running the usage report for Amazon S3, too. However, if you use Amazon S3 for purposes other than as an origin for your CloudFront distributions, it might not be clear what portion applies to your CloudFront usage.

> **Tip**
> For detailed information about every request that CloudFront receives for your objects, turn on CloudFront access logs for your distribution. For more information, see Access Logs (p. 182).

To download the usage report for CloudFront or Amazon S3

1. Sign in to the AWS Management Console at https://console.aws.amazon.com/console/home.
2. In the title bar, click your IAM user name, and click **Billing & Cost Management**.
3. In the navigation pane, click **Reports**.
4. Under **AWS Usage Report**, click **AWS Usage Report**.
5. In the **Service** list, click **CloudFront** or **Amazon Simple Storage Service**.
6. Select the applicable settings:

- **Usage Types** — For a detailed explanation of CloudFront usage types, see the section called "Interpreting Your AWS Bill and the CloudFront Usage Report" (p. 14).

  For Amazon S3, select **All Usage Types**.
- **Operation** — Select **All Operations**.
- **Time Period** — Select the time period that you want the report to cover.
- **Report Granularity** — Select whether you want the report to include subtotals by the hour, by the day, or by the month.

7. Click the download button for the desired format.
8. Follow the on-screen prompts to view or save the report.

CloudFront Usage Charts

The Amazon CloudFront console can display a graphical representation of your CloudFront usage that is based on a subset of the usage report data. You can display charts for a specified date range in the last 60 days, with data points every hour or every day. You can usually view data about requests that CloudFront received as recently as four hours ago, but data can occasionally be delayed by as much as 24 hours.

For more information, see How the Usage Charts Are Related to Data in the CloudFront Usage Report (p. 12).

To display CloudFront usage charts

1. Sign in to the AWS Management Console and open the Amazon CloudFront console at https://console.aws.amazon.com/cloudfront/.
2. In **Navigation** pane, click **Reports & Analytics**.
3. In the **Reports and Analytics** pane, for **From** and **To**, select the date range for which you want to display usage charts. Available ranges depend on the value that you select for **Granularity**:

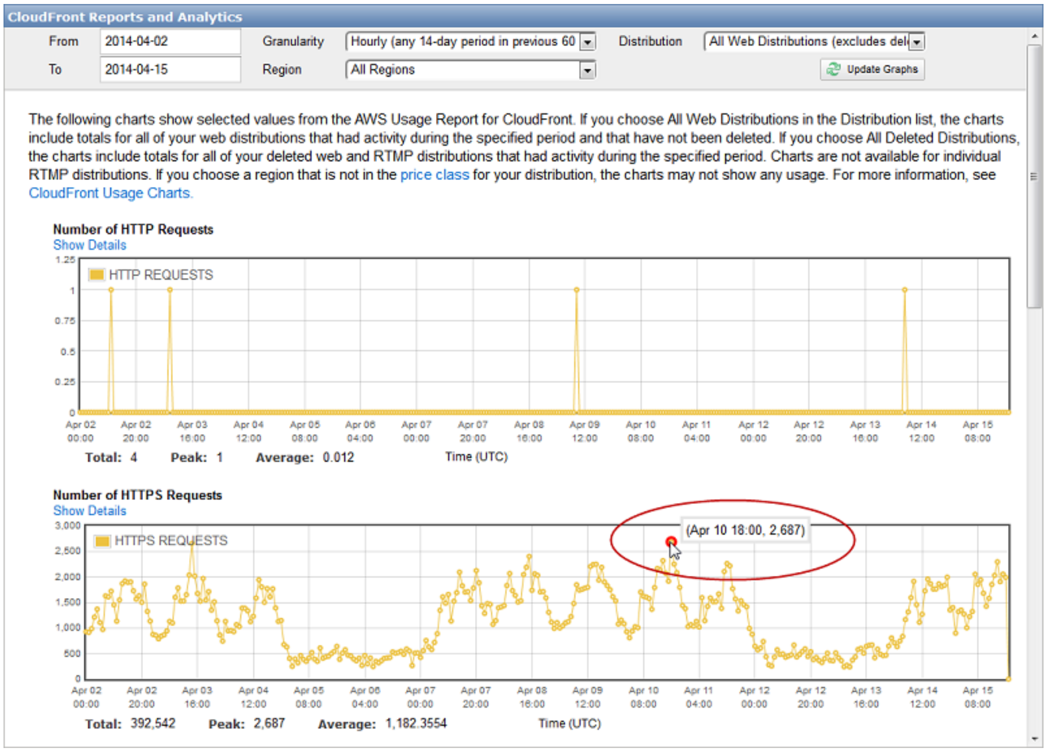
   - **Daily** — To display charts with one data point per day, select any date range in the previous 60 days.
   - **Hourly** — To display charts with one data point every hour, select any date range of up to 14 days within the previous 60 days.

   Dates and times are computed based on Coordinated Universal Time (UTC).
4. For **Granularity**, specify whether to display one data point per day or one data point per hour in the charts. If you specify a date range greater than 14 days, the option to specify one data point per hour is not available.
5. For **Region**, choose the CloudFront billing region that has the data you want to view, or choose **All Regions**. Usage charts include data for requests that CloudFront processes in edge locations in the specified region. The region where CloudFront processes requests might or might not correspond with the location of your users.

   Select only regions that are included in the price class for your distribution; otherwise, the usage charts probably won't contain any data. For example, if you chose Price Class 200 for your distribution, the South America and Australia billing regions are not included, so CloudFront generally won't process your requests from those regions. For more information about price classes, see Choosing the Price Class for a CloudFront Distribution (p. 32).
6. In the **Distribution** list, select the distributions for which you want to display data in the usage charts:

- **An individual web distribution** — The charts display data for the selected CloudFront distribution. The **Distribution** list displays the distribution ID and alternate domain names (CNAMEs) for the distribution, if any. If a distribution has no alternate domain names, the list includes origin domain names for the distribution.
- **All Web Distributions (excludes deleted)** — The charts display summed data for all web distributions that are associated with the current AWS account, excluding web distributions that you have deleted.
- **All Deleted Distributions** — The charts display summed data for all web and RTMP distributions that are associated with the current AWS account and that were deleted in the last 60 days.

7. Click **Update Graphs**.
8. To view data for a daily or hourly data point within a chart, move your mouse pointer over the data point.



CloudFront Reports and Analytics

From 2014-04-02
To 2014-04-15
Granularity Hourly (any 14-day period in previous 60
Region All Regions
Distribution All Web Distributions (excludes del

The following charts show selected values from the AWS Usage Report for CloudFront. If you choose All Web Distributions in the Distribution list, the charts include totals for all of your web distributions that had activity during the specified period and that have not been deleted. If you choose All Deleted Distributions, the charts include totals for all of your deleted web and RTMP distributions that had activity during the specified period. Charts are not available for individual RTMP distributions. If you choose a region that is not in the price class for your distribution, the charts may not show any usage. For more information, see CloudFront Usage Charts.

Number of HTTP Requests

Total: 4 Peak: 1 Average: 0.012

Number of HTTPS Requests

Total: 392,542 Peak: 2,687 Average: 1,182.3554

9. For charts that show data transferred, note that you can change the vertical scale to gigabytes, megabytes, or kilobytes for each chart.

How the Usage Charts Are Related to Data in the CloudFront Usage Report

Here's how the usage charts map to values in the CloudFront usage report.

Chart Name	Values in the Usage Type Column in the CloudFront Usage Report
Number of HTTP Requests	• *region*-**Requests-HTTP-Static:** Number of HTTP GET and HEAD requests served for objects with TTL ≥ 3600 seconds • *region*-**Requests-HTTP-Dynamic:** Number of HTTP GET and HEAD requests served for objects with TTL < 3600 seconds • *region*-**Requests-HTTP-Proxy:** Number of HTTP DELETE, OPTIONS, PATCH, POST, and PUT requests that CloudFront forwards to your origin
Number of HTTPS Requests	• *region*-**Requests-HTTPS-Static:** Number of HTTPS GET and HEAD requests served for objects with TTL ≥ 3600 seconds • *region*-**Requests-HTTPS-Dynamic:** Number of HTTPS GET and HEAD requests served for objects with TTL < 3600 seconds • *region*-**Requests-HTTPS-Proxy:** Number of HTTPS DELETE, OPTIONS, PATCH, POST, and PUT requests that CloudFront forwards to your origin
Data Transferred over HTTP	• *region*-**Out-Bytes-HTTP-Static:** Bytes served via HTTP for objects with TTL ≥ 3600 seconds • *region*-**Out-Bytes-HTTP-Dynamic:** Bytes served via HTTP for objects with TTL < 3600 seconds • *region*-**Out-Bytes-HTTP-Proxy:** Bytes returned from CloudFront to viewers via HTTP in response to DELETE, OPTIONS, PATCH, POST, and PUT requests. • *region*-**Out-OBytes-HTTP-Proxy:** Total bytes transferred via HTTP from CloudFront edge locations to your origin in response to DELETE, OPTIONS, PATCH, POST, and PUT requests.
Data Transferred over HTTPS	• *region*-**Out-Bytes-HTTPS-Static:** Bytes served via HTTPS for objects with TTL ≥ 3600 seconds • *region*-**Out-Bytes-HTTPS-Dynamic:** Bytes served via HTTPS for objects with TTL < 3600 seconds • *region*-**Out-Bytes-HTTPS-Proxy:** Bytes returned from CloudFront to viewers via HTTPS in response to DELETE, OPTIONS, PATCH, POST, and PUT requests. • *region*-**Out-OBytes-HTTPS-Proxy:** Total bytes transferred via HTTPS from CloudFront edge locations to your origin in response to DELETE, OPTIONS, PATCH, POST, and PUT requests.

Chart Name	Values in the Usage Type Column in the CloudFront Usage Report
Data Transferred from CloudFront Edge Locations to Your Users	• *region*-**Out-Bytes-HTTP-Static:** Bytes served via HTTP for objects with TTL ≥ 3600 seconds • *region*-**Out-Bytes-HTTPS-Static:** Bytes served via HTTPS for objects with TTL ≥ 3600 seconds • *region*-**Out-Bytes-HTTP-Dynamic:** Bytes served via HTTP for objects with TTL < 3600 seconds • *region*-**Out-Bytes-HTTPS-Dynamic:** Bytes served via HTTPS for objects with TTL < 3600 seconds • *region*-**Out-Bytes-HTTP-Proxy:** Bytes returned from CloudFront to viewers via HTTP in response to `DELETE`, `OPTIONS`, `PATCH`, `POST`, and `PUT` requests. • *region*-**Out-Bytes-HTTPS-Proxy:** Bytes returned from CloudFront to viewers via HTTPS in response to `DELETE`, `OPTIONS`, `PATCH`, `POST`, and `PUT` requests.
Data Transferred from CloudFront to Your Origin	• *region*-**Out-OBytes-HTTP-Proxy:** Total bytes transferred via HTTP from CloudFront edge locations to your origin in response to `DELETE`, `OPTIONS`, `PATCH`, `POST`, and `PUT` requests. • *region*-**Out-OBytes-HTTPS-Proxy:** Total bytes transferred via HTTPS from CloudFront edge locations to your origin in response to `DELETE`, `OPTIONS`, `PATCH`, `POST`, and `PUT` requests.

Interpreting Your AWS Bill and the CloudFront Usage Report

Your AWS bill for CloudFront service includes codes and abbreviations that might not be immediately obvious. The first column in the following table lists items that appear in your bill and explains what each means.

In addition, you can get an AWS usage report for CloudFront that contains more detail than the AWS bill for CloudFront. The second column in the table lists items that appear in the usage report and shows the correlation between bill items and usage report items.

Most codes in both columns include a two-letter abbreviation that indicates the location of the activity. In the following table, *region* in a code is replaced by one of the following two-letter abbreviations in your AWS bill and in the usage report:

- **AP:** Hong Kong, Philippines, South Korea, Singapore, and Taiwan (Asia Pacific)
- **AU:** Australia
- **EU:** Europe
- **IN:** India
- **JP:** Japan
- **SA:** South America
- **US:** United States

For more information about pricing by region, see Amazon CloudFront Pricing.

Note
This table doesn't include charges for transferring your objects from an Amazon S3 bucket to CloudFront edge locations. These charges, if any, appear in the **AWS Data Transfer** portion of your AWS bill.

Items in Your CloudFront Bill	Values in the Usage Type Column in the CloudFront Usage Report
*region***-DataTransfer-Out-Bytes** Sum of bytes that CloudFront served for web and RTMP distributions: • **Web distributions:** Total bytes served from CloudFront edge locations in *region* in response to user `GET` and `HEAD` requests • **RTMP distributions:** Total bytes transferred from CloudFront edge locations in *region* to end users	**Web distributions:** • *region***-Out-Bytes-HTTP-Static:** Bytes served via HTTP for objects with TTL ≥ 3600 seconds • *region***-Out-Bytes-HTTPS-Static:** Bytes served via HTTPS for objects with TTL ≥ 3600 seconds • *region***-Out-Bytes-HTTP-Dynamic:** Bytes served via HTTP for objects with TTL < 3600 seconds • *region***-Out-Bytes-HTTPS-Dynamic:** Bytes served via HTTPS for objects with TTL < 3600 seconds • *region***-Out-Bytes-HTTP-Proxy:** Bytes returned from CloudFront to viewers via HTTP in response to `DELETE`, `OPTIONS`, `PATCH`, `POST`, and `PUT` requests. • *region***-Out-Bytes-HTTPS-Proxy:** Bytes returned from CloudFront to viewers via HTTPS in response to `DELETE`, `OPTIONS`, `PATCH`, `POST`, and `PUT` requests. **RTMP distributions:** • *region***-FMS-Out-Bytes**
*region***-DataTransfer-Out-OBytes** **Web distributions only:** Total bytes transferred from CloudFront edge locations to your origin in response to `DELETE`, `OPTIONS`, `PATCH`, `POST`, and `PUT` requests.	*region***-Out-OBytes-HTTP-Proxy** Total bytes transferred via HTTP from CloudFront edge locations to your origin in response to `DELETE`, `OPTIONS`, `PATCH`, `POST`, and `PUT` requests. *region***-Out-OBytes-HTTPS-Proxy** Total bytes transferred via HTTPS from CloudFront edge locations to your origin in response to `DELETE`, `OPTIONS`, `PATCH`, `POST`, and `PUT` requests.
*region***-Requests-Tier1** **Web distributions only:** Number of HTTP `GET` and `HEAD` requests	*region***-Requests-HTTP-Static** Number of HTTP `GET` and `HEAD` requests served for objects with TTL ≥ 3600 seconds *region***-Requests-HTTP-Dynamic** Number of HTTP `GET` and `HEAD` requests served for objects with TTL < 3600 seconds

Items in Your CloudFront Bill	Values in the Usage Type Column in the CloudFront Usage Report
region-**Requests-Tier2-HTTPS** **Web distributions only:** Number of HTTPS `GET` and `HEAD` requests	*region*-**Requests-HTTPS-Static** Number of HTTPS `GET` and `HEAD` requests served for objects with TTL ≥ 3600 seconds *region*-**Requests-HTTPS-Dynamic** Number of HTTPS `GET` and `HEAD` requests served for objects with TTL < 3600 seconds
region-**Requests-HTTP-Proxy** **Web distributions only:** Number of HTTP `DELETE`, `OPTIONS`, `PATCH`, `POST`, and `PUT` requests that CloudFront forwards to your origin	*region*-**Requests-HTTP-Proxy** Same as the corresponding item in your CloudFront bill
region-**Requests-HTTPS-Proxy** **Web distributions only:** Number of HTTPS `DELETE`, `OPTIONS`, `PATCH`, `POST`, and `PUT` requests that CloudFront forwards to your origin	*region*-**Requests-HTTPS-Proxy** Same as the corresponding item in your CloudFront bill
Invalidations **Web distributions only:** The charge for invalidating objects (removing the objects from CloudFront edge locations); for more information, see Paying for Object Invalidation (p. 91)	**Invalidations** Same as the corresponding item in your CloudFront bill
SSL-Cert-Custom **Web distributions only:** The charge for using an SSL certificate with a CloudFront alternate domain name such as example.com instead of using the default CloudFront SSL certificate and the domain name that CloudFront assigned to your distribution	**SSL-Cert-Custom** Same as the corresponding item in your CloudFront bill

Getting Started with CloudFront

The example in this topic gives you a quick overview of how to use CloudFront to:

- Store the original versions of your objects in one Amazon Simple Storage Service (Amazon S3) bucket.
- Distribute download content such as text or graphics.
- Make your objects accessible to everyone.
- Use the CloudFront domain name in URLs for your objects (for example, `http://d111111abcdef8.cloudfront.net/image.jpg`) instead of your own domain name (for example, `http://www.example.com/image.jpg`).
- Keep your objects in CloudFront edge locations for the default duration of 24 hours. (The minimum duration is 0 seconds.)

For information about how to use CloudFront when you want to use other options, see Task List for Creating a Web Distribution (p. 36) or Task List for Streaming Media Files Using RTMP (p. 62).

You only need to perform a few basic steps to start delivering your content using CloudFront. The first step is signing up. After that, you create a CloudFront distribution, and then use the CloudFront domain name to reference content in your web pages or applications.

Step 1: Sign up for Amazon Web Services

If you haven't already done so, sign up for Amazon Web Services at http://aws.amazon.com. Just click **Sign Up Now** and enter any required information.

Step 2: Upload your content to Amazon S3 and grant object permissions

An Amazon S3 bucket is a container that can contain objects or folders. CloudFront can distribute almost any type of object for you using an Amazon S3 bucket as the source, for example, text, images, and videos. You can create multiple buckets, and there is no limit to the amount of data that you can store on Amazon S3.

By default, your Amazon S3 bucket and all of the objects in it are private—only the AWS account that created the bucket has permission to read or write the objects in it. If you want to allow anyone to access the objects in your Amazon S3 bucket using CloudFront URLs, you must grant public read permissions to the objects. (This is one of the most common mistakes when working with CloudFront and Amazon S3. You must explicitly grant privileges to each object in an Amazon S3 bucket.)

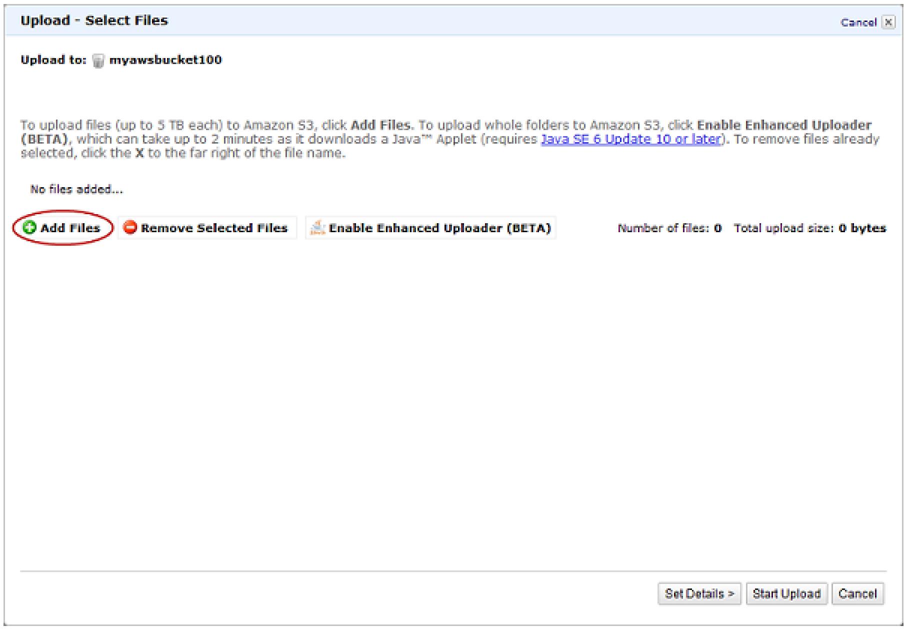
> **Note**
>
> If you want to restrict who can download your content, you can use the CloudFront private content feature. For more information about distributing private content, see Serving Private Content through CloudFront (p. 118).

To upload your content to Amazon S3 and grant read permission to everyone

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In the Amazon S3 console, click **Create Bucket**.
3. In the **Create Bucket** dialog, enter a bucket name.

   > **Important**
   >
   > For your bucket to work with CloudFront, the name must conform to DNS naming requirements. For more information, go to Bucket Restrictions and Limitations in the *Amazon Simple Storage Service Developer Guide*.

4. Select a region for your bucket. By default, Amazon S3 creates buckets in the US-Standard region. We recommend that you choose a region close to you to optimize latency, minimize costs, or to address regulatory requirements.
5. Click **Create**.
6. Select your bucket in the **Buckets** pane, and click **Upload**.
7. On the **Upload - Select Files** page, click **Add Files**, and choose the files that you want to upload.



Upload - Select Files

Upload to: myawsbucket100

To upload files (up to 5 TB each) to Amazon S3, click **Add Files.** To upload whole folders to Amazon S3, click **Enable Enhanced Uploader (BETA)**, which can take up to 2 minutes as it downloads a Java™ Applet (requires Java SE 6 Update 10 or later). To remove files already selected, click the **X** to the far right of the file name.

No files added...

Number of files: **0** Total upload size: **0 bytes**

8. Enable public read privileges for each object that you upload to your Amazon S3 bucket.

   a. Click **Set Details**.
   b. On the **Set Details** page, click **Set Permissions**.
   c. On the **Set Permissions** page, click **Make everything public**.

9. Click **Start Upload**.

   After the upload completes, you can navigate to this item by its URL. In the case of the previous example, the URL would be:

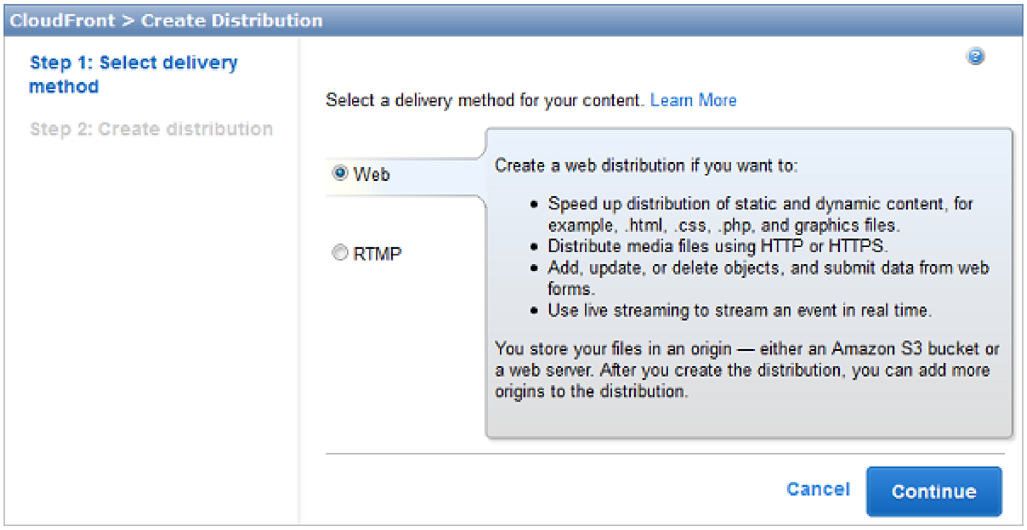
   ```
   http://s3.amazonaws.com/example-myawsbucket/filename
   ```

   Use your Amazon S3 URL to verify that your content is publicly accessible, but remember that this is not the URL you will use when you are ready to distribute your content.

Step 3: Create a CloudFront Web Distribution

To create a CloudFront web distribution

1. Open the Amazon CloudFront console at https://console.aws.amazon.com/cloudfront/.
2. Click **Create Distribution**.
3. On the first page of the **Create Distribution Wizard**, accept the default selection, **Web**, and click **Continue**.



CloudFront > Create Distribution

Step 1: Select delivery method

Step 2: Create distribution

Select a delivery method for your content. Learn More

● Web

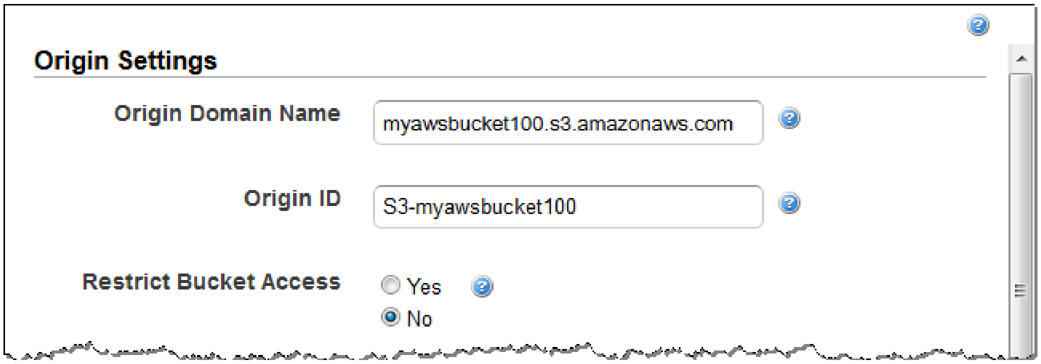
○ RTMP

Create a web distribution if you want to:

- Speed up distribution of static and dynamic content, for example, .html, .css, .php, and graphics files.
- Distribute media files using HTTP or HTTPS.
- Add, update, or delete objects, and submit data from web forms.
- Use live streaming to stream an event in real time.

You store your files in an origin — either an Amazon S3 bucket or a web server. After you create the distribution, you can add more origins to the distribution.

4. Select the Amazon S3 bucket that you created earlier. For **Origin ID** and **Restrict Bucket Access**, accept the default values.



Origin Settings

Origin Domain Name: myawsbucket100.s3.amazonaws.com

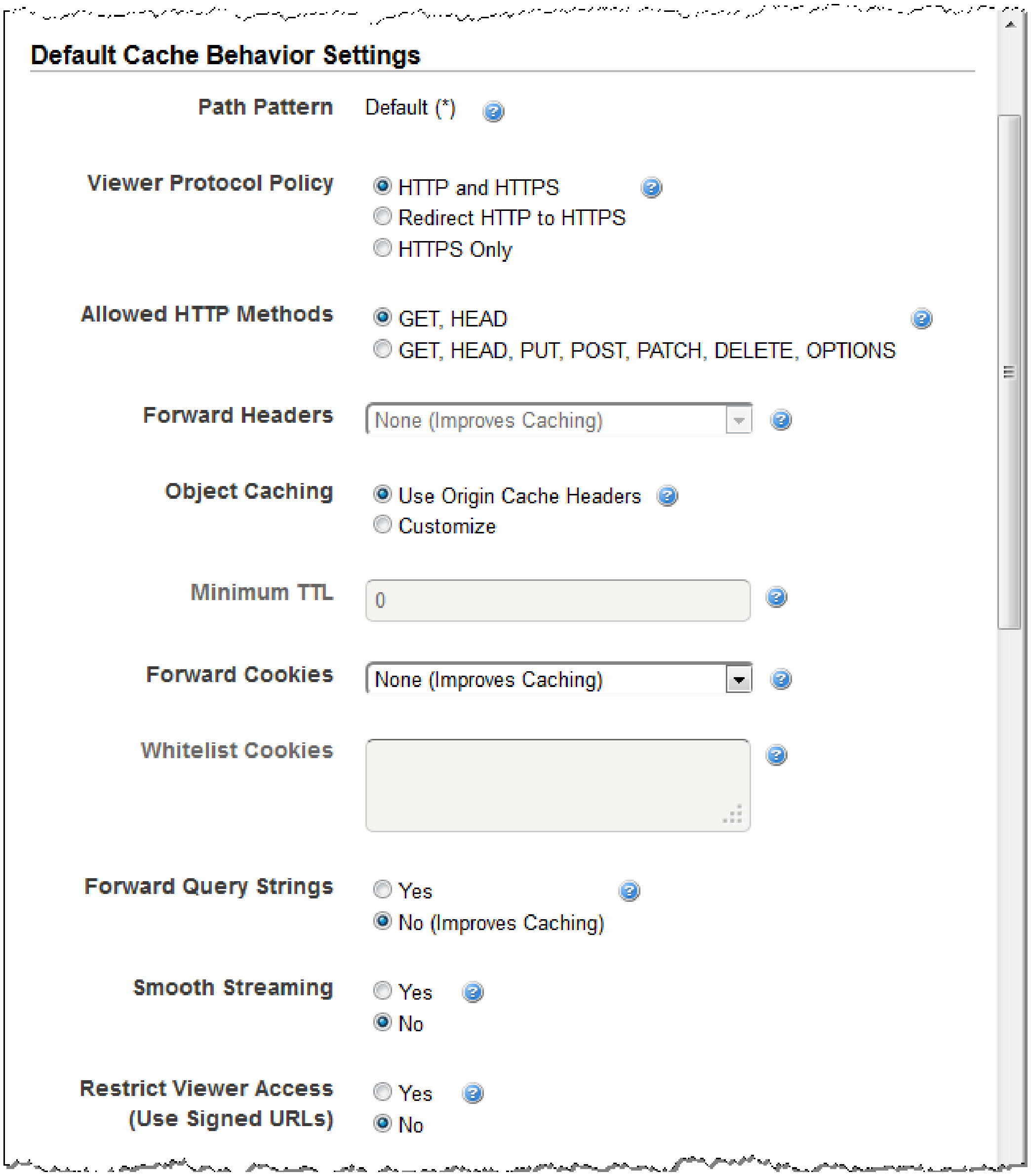
Origin ID: S3-myawsbucket100

Restrict Bucket Access: ○ Yes ● No

5. Under **Default Cache Behavior Settings**, accept the default values, and CloudFront will:

- Forward all requests that use the CloudFront URL for your distribution (for example, `http://d111111abcdef8.cloudfront.net/image.jpg`) to the Amazon S3 bucket that you specified in Step 4.
- Allow end users to use either HTTP or HTTPS to access your objects.
- Respond to requests for your objects.
- Cache your objects at CloudFront edge locations for 24 hours.
- Forward only the default request headers to your origin and not cache your objects based on the values in the headers.
- Exclude cookies and query string parameters, if any, when forwarding requests for objects to your origin. (Amazon S3 doesn't process cookies and processes only a limited set of query string parameters.)
- Not be configured to distribute media files in the Microsoft Smooth Streaming format.
- Allow everyone to view your content.

For more information about cache behavior options, see Cache Behavior Settings (p. 44).



Default Cache Behavior Settings

Path Pattern	Default (*)
Viewer Protocol Policy	● HTTP and HTTPS ○ Redirect HTTP to HTTPS ○ HTTPS Only
Allowed HTTP Methods	● GET, HEAD ○ GET, HEAD, PUT, POST, PATCH, DELETE, OPTIONS
Forward Headers	None (Improves Caching)
Object Caching	● Use Origin Cache Headers ○ Customize
Minimum TTL	0
Forward Cookies	None (Improves Caching)
Whitelist Cookies	
Forward Query Strings	○ Yes ● No (Improves Caching)
Smooth Streaming	○ Yes ● No
Restrict Viewer Access (Use Signed URLs)	○ Yes ● No

6. Under **Distribution Details**, enter the applicable values:

   - **Price Class:** Select the price class that corresponds with the maximum price that you want to pay for CloudFront service. By default, CloudFront serves your objects from edge locations in all CloudFront regions.

     For more information about price classes and about how your choice of price class affects CloudFront performance for your distribution, go to Choosing the Price Class for a CloudFront Distribution (p. 32). For information about CloudFront pricing, including how price classes map to CloudFront regions, go to Amazon CloudFront Pricing.
   - **Alternate Domain Names (CNAMEs):** (Optional) Specify one or more domain names that you want to use for URLs for your objects instead of the domain name that CloudFront assigns when you create your distribution. For example, if you want the URL for the object:

     ```
     /images/image.jpg
     ```

to look like this:

```
http://www.example.com/images/image.jpg
```

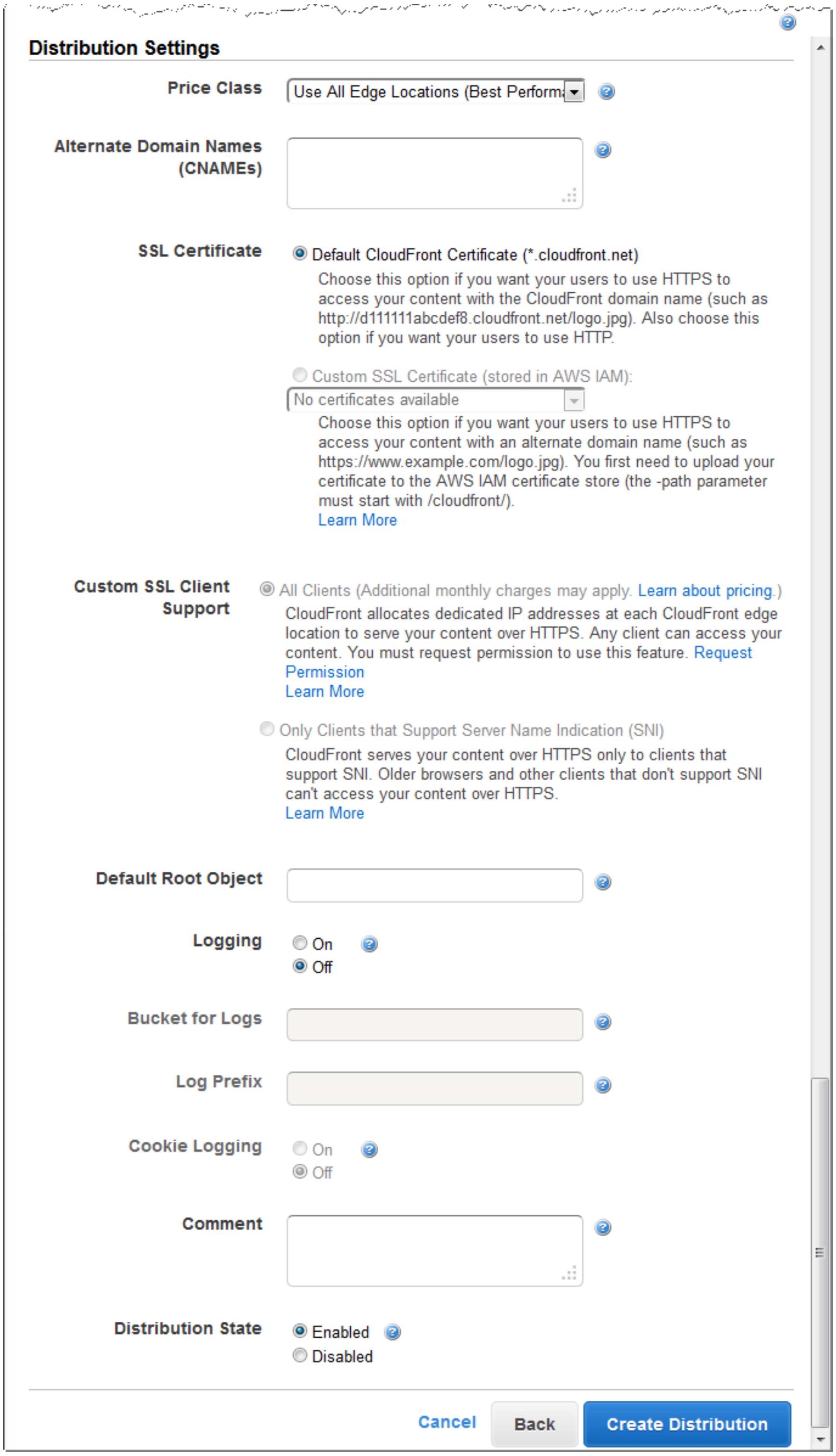
instead of like this:

```
http://d111111abcdef8.cloudfront.net/images/image.jpg
```

you would create a CNAME for `www.example.com`. You can create up to 10 CNAMEs per distribution.

> **Important**
> If you add a CNAME for `www.example.com` to your distribution, you also need to create (or update) a CNAME record with your DNS service to route queries for `www.example.com` to `d111111abcdef8.cloudfront.net`. You must have permission to create a CNAME record with the DNS service provider for the domain. Typically, this means that you own the domain, but you may also be developing an application for the domain owner. For more information about CNAMEs, see Using Alternate Domain Names (CNAMEs) (p. 29).

- **SSL Certificate:** Accept the default value, **Default CloudFront Certificate**.
- **Clients Supported:** When you choose the default value for **SSL Certificate**, this setting is unavailable.
- **Default Root Object:** (Optional) The object that you want CloudFront to request from your origin (for example, `index.html`) when a viewer requests the root URL of your distribution (`http://www.example.com/`) instead of an object in your distribution (`http://www.example.com/product-description.html`). Specifying a default root object avoids exposing the contents of your distribution.
- **Logging:** (Optional) If you want CloudFront to log information about each request for an object and store the log files in an Amazon S3 bucket, select **On**, and specify the bucket and an optional prefix for the names of the log files. There is no extra charge to enable logging, but you accrue the usual Amazon S3 charges for storing and accessing the files. CloudFront doesn't delete the logs automatically, but you can delete them at any time.
- **Cookie Logging:** In this example, we're using Amazon S3 as the origin for your objects, and Amazon S3 doesn't process cookies, so we recommend that you select **Off** for the value of **Cookie Logging**.
- **Comment:** (Optional) Enter any comments that you want to save with the distribution.
- **Distribution State:** Select **Enabled** if you want CloudFront to begin processing requests as soon as the distribution is created, or select **Disabled** if you do not want CloudFront to begin processing requests after the distribution is created.



Distribution Settings

Price Class: Use All Edge Locations (Best Perform▾

Alternate Domain Names (CNAMEs):

SSL Certificate:

● Default CloudFront Certificate (*.cloudfront.net)

Choose this option if you want your users to use HTTPS to access your content with the CloudFront domain name (such as http://d111111abcdef8.cloudfront.net/logo.jpg). Also choose this option if you want your users to use HTTP.

○ Custom SSL Certificate (stored in AWS IAM):

No certificates available ▾

Choose this option if you want your users to use HTTPS to access your content with an alternate domain name (such as https://www.example.com/logo.jpg). You first need to upload your certificate to the AWS IAM certificate store (the -path parameter must start with /cloudfront/).

Learn More

Custom SSL Client Support:

● All Clients (Additional monthly charges may apply. Learn about pricing.)

CloudFront allocates dedicated IP addresses at each CloudFront edge location to serve your content over HTTPS. Any client can access your content. You must request permission to use this feature. Request Permission

Learn More

○ Only Clients that Support Server Name Indication (SNI)

CloudFront serves your content over HTTPS only to clients that support SNI. Older browsers and other clients that don't support SNI can't access your content over HTTPS.

Learn More

Default Root Object:

Logging: ○ On ● Off

Bucket for Logs:

Log Prefix:

Cookie Logging: ○ On ● Off

Comment:

Distribution State: ● Enabled ○ Disabled

7. Click **Create Distribution**.

8. After CloudFront has created your distribution, the value of the **Status** column for your distribution will change from **InProgress** to **Deployed**. If you chose to enable the distribution, it will then be ready to process requests. This should take less than 15 minutes.

   The domain name that CloudFront assigns to your distribution appears in the list of distributions. (It also appears on the **General** tab for a selected distribution.)

Step 4: Test your links

After you've created your distribution, CloudFront knows where your Amazon S3 origin server is, and you know the domain name associated with the distribution. You can create a link to your Amazon S3 bucket content with that domain name, and have CloudFront serve it.

> **Note**
> You must wait until the status of your distribution changes to **Deployed** before testing your links.

To link to your objects

1. Copy the following HTML into a new file:

   - Replace <domain name> with the domain name that CloudFront assigned to your distribution.
   - Replace <object name> with the name of a file in your Amazon S3 bucket.

   ```
   <html>
   <head>My CloudFront Test</head>
   <body>
   <p>My text content goes here.</p>
   <p> <img src="http://<domain name>/<object name>" alt="my test image" />
   </body>
   </html>
   ```

   For example, if your domain name was `d111111abcdef8.cloudfront.net` and your object was `image.jpg`, the URL for the link would be:

   `http://d111111abcdef8.cloudfront.net/image.jpg`.

   If your object is in a folder within your bucket, include the folder in the URL. For example, if image.jpg is located in an images folder, then the URL would be:

   http://d111111abcdef8.cloudfront.net/images/image.jpg
2. Save the text in a file that has a .html filename extension.
3. Open your web page in a browser to ensure that you can see your content. If you cannot see the content, confirm that you have performed all of the steps correctly. You can also see the tips in Troubleshooting (p. 198).

The browser returns your page with the embedded image file, served from the edge location that CloudFront determined was appropriate to serve the object.

For more information on using CloudFront, go to Amazon CloudFront Resources (p. 283).

Working with Distributions

Topics

- Overview of Web and RTMP Distributions (p. 26)
- Creating Web and RTMP Distributions (p. 27)
- Listing, Viewing, and Updating CloudFront Distributions (p. 27)
- Deleting a Distribution (p. 28)
- Using Alternate Domain Names (CNAMEs) (p. 29)
- Choosing the Price Class for a CloudFront Distribution (p. 32)
- Using CloudFront with Amazon S3 (p. 33)
- Changes to the CloudFront API (p. 34)

The following table lists the actions you can perform on a distribution and provides links to the corresponding documentation on how to perform the actions using the CloudFront console and the CloudFront API.

Action	Using the CloudFront Console	Using the CloudFront API: Web Distributions	Using the CloudFront API: RTMP Distributions
Create a distribution	**Web Distributions:** See Task List for Creating a Web Distribution (p. 36) **RTMP Distributions:** See Task List for Streaming Media Files Using RTMP (p. 62)	Go to POST Distribution	Go to POST Streaming Distribution
List your distributions	See Listing, Viewing, and Updating CloudFront Distributions (p. 27)	Go to GET Distribution List	Go to GET Streaming Distribution List
Get all information about a distribution	See Listing, Viewing, and Updating CloudFront Distributions (p. 27)	Go to GET Distribution	Go to GET Streaming Distribution

Action	Using the CloudFront Console	Using the CloudFront API: Web Distributions	Using the CloudFront API: RTMP Distributions
Get the distribution configuration	See Listing, Viewing, and Updating CloudFront Distributions (p. 27)	Go to GET Distribution Config	Go to GET Streaming Distribution Config
Update a distribution	See Listing, Viewing, and Updating CloudFront Distributions (p. 27)	Go to PUT Distribution Config	Go to PUT Streaming Distribution Config
Delete a distribution	See Deleting a Distribution (p. 28)	Go to DELETE Distribution	Go to DELETE Streaming Distribution

Overview of Web and RTMP Distributions

When you want to use CloudFront to distribute your content, you create a distribution and specify configuration settings such as:

- Your origin, which is the Amazon S3 bucket or HTTP server from which CloudFront gets the files that it distributes. You can specify any combination of up to 10 Amazon S3 buckets and/or HTTP servers as your origins.
- Whether you want the files to be available to everyone or you want to restrict access to selected users.
- Whether you want CloudFront to require users to use HTTPS to access your content.
- Whether you want CloudFront to forward cookies and/or query strings to your origin.
- Whether you want CloudFront to prevent users in selected countries from accessing your content.
- Whether you want CloudFront to create access logs.

For the current limit on the number of web and RTMP distributions that you can create for each AWS account, see Amazon CloudFront Limits in the *Amazon Web Services General Reference*. To request a higher limit, go to https://aws.amazon.com/support/createCase?type=service_limit_increase&serviceLimitIncreaseType=cloudfront-distributions.

The number of files that you can serve per distribution is unlimited.

Web Distributions

You can use web distributions to serve the following content over HTTP or HTTPS:

- Static and dynamic download content, for example, .html, .css, .php, and image files, using HTTP or HTTPS.
- Multimedia content on demand using progressive download and Apple HTTP Live Streaming (HLS). For more information, see the applicable topic in Working with Web Distributions (p. 36).

  You can't serve Adobe Flash multimedia content over HTTP or HTTPS, but you can serve it using a CloudFront RTMP distribution. See RTMP Distributions (p. 27) below.
- A live event, such as a meeting, conference, or concert, in real time. For live streaming, you create the distribution automatically by using an AWS CloudFormation stack. For more information, see the applicable live-streaming tutorial in CloudFront Tutorials (p. 202).

For web distributions, your origin can be either an Amazon S3 bucket or an HTTP server, for example, a web server. For more information about how web distributions work, including the values that you specify when you create a web distribution, see Working with Web Distributions (p. 36). For information about creating a web distribution, see Task List for Creating a Web Distribution (p. 36).

RTMP Distributions

RTMP distributions stream media files using Adobe Media Server and the Adobe Real-Time Messaging Protocol (RTMP). An RTMP distribution must use an Amazon S3 bucket as the origin.

For information about the values you specify when you create an RTMP distribution, see Working with RTMP Distributions (p. 60). For information about creating an RTMP distribution, see Task List for Streaming Media Files Using RTMP (p. 62).

Creating Web and RTMP Distributions

Web distributions: For information about creating web distributions using the CloudFront console, see Task List for Creating a Web Distribution (p. 36). For information about creating web distributions using the CloudFront API, see POST Distribution in the *Amazon CloudFront API Reference*.

RTMP distributions: For information about creating RTMP distributions using the CloudFront console, see Task List for Streaming Media Files Using RTMP (p. 62). For information about creating RTMP distributions using the CloudFront API, see POST Streaming Distribution in the *Amazon CloudFront API Reference*.

Listing, Viewing, and Updating CloudFront Distributions

You can use the CloudFront console to list the CloudFront distributions that are associated with your AWS account, view the settings for a distribution, and update most settings.

When you save changes to your distribution configuration, CloudFront starts to propagate the changes to all edge locations. Until your configuration is updated in an edge location, CloudFront continues to serve your content from that location based on the previous configuration. After your configuration is updated in an edge location, CloudFront immediately starts to serve your content from that location based on the new configuration.

Your changes don't propagate to every edge location instantaneously; propagation to all edge locations can take 15 minutes. When propagation is complete, the status of your distribution changes from **InProgress** to **Deployed**. While CloudFront is propagating your changes to edge locations, we cannot determine whether a given edge location is serving your content based on the previous configuration or the new configuration.

To List, View, and Update CloudFront Distributions Using the CloudFront Console

1. Sign in to the AWS Management Console and open the Amazon CloudFront console at https://console.aws.amazon.com/cloudfront/.
2. In the top pane of the CloudFront console, select the distribution that you want to view or update.

   Note
   The top pane lists all of the distributions that are associated with the AWS account that you used when you signed in to the CloudFront console.

3. To view or edit RTMP distribution settings, skip to Step 4.

   To view or edit settings for a web distribution, perform the following steps.

   a. In the **Distribution Settings** pane, click the tab for the settings that you want to change: **General**, **Origins**, or **Behaviors**.
   b. For general settings, click **Edit**.

      For origins or cache behaviors, click the origin or cache behavior, and click **Edit**.
   c. Enter or update the applicable values. For information about the fields, see the following topics:

      - **General settings:** Distribution Details (p. 50)
      - **Origin settings:** Origin Settings (p. 42)
      - **Cache behavior settings:** Cache Behavior Settings (p. 44)

   d. Click **Yes, Edit**.

4. To edit or view settings for an RTMP distribution:

   a. In the **Distribution Details** pane, click **Edit**.
   b. Enter or update the applicable values. For information about the fields, see Values that You Specify When You Create or Update an RTMP Distribution (p. 63).
   c. Click **Yes, Edit**.

Deleting a Distribution

If you no longer want to use a distribution, use the following procedure to delete it using the CloudFront console.

You can also delete a distribution using the CloudFront API:

- To delete a web distribution, use the `DELETE Distribution` API action. For more information, go to DELETE Distribution in the *Amazon CloudFront API Reference*.
- To delete an RTMP distribution, use the `DELETE Streaming Distribution` API action. For more information, go to DELETE Streaming Distribution in the *Amazon CloudFront API Reference*.

> **Note**
> CloudFront lets you create a combined total of up to 100 web and RTMP distributions for an AWS account.

To Delete a CloudFront Distribution Using the CloudFront Console

1. Sign in to the AWS Management Console and open the Amazon CloudFront console at https://console.aws.amazon.com/cloudfront/.
2. In the right pane of the CloudFront console, find the distribution that you want to delete.
3. If the value of the **State** column is **Disabled**, skip to Step 7.

   If the value of **State** is **Enabled** and the value of **Status** is **Deployed**, continue with Step 4 to disable the distribution before deleting it.

If the value of **State** is **Enabled** and the value of **Status** is **InProgress**, wait until **Status** changes to **Deployed**. Then continue with Step 4 to disable the distribution before deleting it.

4. In the right pane of the CloudFront console, check the check box for the distribution that you want to delete.
5. Click **Disabled** to disable the distribution, and click **Yes, Disable** to confirm. Then click **Close**.
6. The value of the **State** column immediately changes to **Disabled**. Wait until the value of the **Status** column changes to **Deployed**.
7. Check the check box for the distribution that you want to delete.
8. Click **Delete**, and click **Yes, Delete** to confirm. Then click **Close**.

Using Alternate Domain Names (CNAMEs)

Topics

- Using the * Wildcard in Alternate Domain Names (p. 29)
- Restrictions on Using Alternate Domain Names (p. 30)
- Adding an Alternate Domain Name (p. 30)

In CloudFront, an alternate domain name, also known as a CNAME, lets you use your own domain name (for example, `www.example.com`) for links to your objects instead of using the domain name that CloudFront assigns to your distribution. Both web and RTMP distributions support alternate domain names.

When you create a distribution, CloudFront returns a domain name for the distribution, for example:

```
d111111abcdef8.cloudfront.net
```

When you use the CloudFront domain name for your objects, the URL for an object called `/images/image.jpg` is:

```
http://d111111abcdef8.cloudfront.net/images/image.jpg
```

If you want to use your own domain name, such as `www.example.com`, instead of the `cloudfront.net` domain name that CloudFront assigned to your distribution, you can add an alternate domain name to your distribution for `www.example.com`. You can then use the following URL for `/images/image.jpg`:

```
http://www.example.com/images/image.jpg
```

Using the * Wildcard in Alternate Domain Names

When you add alternate domain names, you can use the * wildcard at the beginning of a domain name instead of specifying subdomains individually. For example, with an alternate domain name of `*.example.com`, you can use any domain name that ends with example.com in your object URLs, such as `www.example.com`, `product-name.example.com`, and `marketing.product-name.example.com`. The name of an object is the same regardless of the domain name, for example:

```
www.example.com/images/image.jpg
```

```
product-name.example.com/images/image.jpg
```

```
marketing.product-name.example.com/images/image.jpg
```

The alternate domain name must begin with an asterisk and a dot (`*.`). You *cannot* use a wildcard to replace part of a subdomain name, like this: `*domain.example.com`, and you cannot replace a subdomain in the middle of a domain name, like this: `subdomain.*.example.com`.

A wildcard alternate domain name can overlap with another alternate domain name as long as they're both in the same CloudFront distribution. For example, you can use both `www.example.com` and `*.example.com` as alternate domain names, but they must be in the same distribution.

Restrictions on Using Alternate Domain Names

Note the following restrictions on using alternate domain names:

- For the current limit on the number of alternate domain names that you can add to a distribution, see Amazon CloudFront Limits in the *Amazon Web Services General Reference*. To request a higher limit, go to https://aws.amazon.com/support/createCase?type=service_limit_increase&serviceLimitIncreaseType=cloudfront-distributions.
- You must have permission to create a CNAME record with the DNS service provider for the domain. Typically, this means that you own the domain, but you may also be developing an application for the domain owner.
- You cannot add an alternate domain name to a CloudFront distribution if the alternate domain name already exists in another CloudFront distribution, even if your AWS account owns the other distribution.
- The DNS protocol does not allow you to create a CNAME record for the top node of a DNS namespace, also known as the zone apex. For example, if you register the DNS name `example.com`, the zone apex is `example.com`. You cannot create a CNAME record for `example.com`, but you can create CNAME records for `www.example.com`, `newproduct.example.com`, and so on.

  If you're using Amazon Route 53 as your DNS service, you can create an alias resource record set instead of a CNAME. With an alias resource record set, you don't pay for Amazon Route 53 queries. In addition, you can create an alias resource record set for a domain name at the zone apex (example.com). For more information, go to Routing Queries to an Amazon CloudFront Distribution in the *Amazon Route 53 Developer Guide*.
- If you want viewers to use HTTPS with an alternate domain names, additional configuration is required. For more information, see Using Alternate Domain Names and HTTPS (p. 171).

Adding an Alternate Domain Name

The following task list describes the process for using the CloudFront console to add an alternate domain name to your distribution so you can use your own domain name in your links instead of the CloudFront domain name that is associated with your distribution.

> **Note**
> If you want viewers to use HTTPS with your alternate domain name, see Using Alternate Domain Names and HTTPS (p. 171).

For information about updating your distribution using the CloudFront API, see Working with Distributions (p. 25).

Process for Adding an Alternate Domain Name Using the CloudFront Console

1. Sign in to the AWS Management Console and open the Amazon CloudFront console at https://console.aws.amazon.com/cloudfront/.
2. In the CloudFront console, use the steps below to update your distribution to include your domain name as an alternate domain name in the **Alternate Domain Names (CNAMEs)** field.

   a. In the top pane of the CloudFront console, select the distribution that you want to update, and click **Distribution Settings**.

   b. On the **General** tab, click **Edit**.

   c. Add the applicable alternate domain names in the **Alternate Domain Names (CNAMEs)** field. Separate domain names with commas or put each one on a new line.

d. **Web distributions only:** For **SSL Certificate**, choose the applicable option:

  - **If you don't want to use SSL:** Click **Default CloudFront Certificate**.
  - **If you do want to use SSL:** Click **Custom SSL Certificate Stored in IAM**, and select a certificate from the list.

    If the desired certificate doesn't appear in the list, review the procedure To use alternate domain names with HTTPS (p. 174) to confirm that you correctly uploaded the certificate to the IAM certificate store.

    If you choose this setting, we recommend that you use only an alternate domain name in your object URLs (https://example.com/logo.jpg). If you use your CloudFront distribution domain name (https://d111111abcdef8.cloudfront.net/logo.jpg) and the viewer supports SNI, then CloudFront behaves normally. However, a viewer that does not support SNI exhibits one of the following behaviors, depending on the value of **Clients Supported**:

    - **All Clients**: If the viewer doesn't support SNI, it displays a warning because the CloudFront domain name doesn't match the domain name in your SSL certificate.
    - **Only Clients that Support Server Name Indication (SNI)**: CloudFront drops the connection with the viewer without returning the object.

e. **Web distributions only:** Choose the applicable option for **Clients Supported**:

  - **All Clients**: CloudFront serves your HTTPS content using dedicated IP addresses. If you select this option, you incur additional charges when you associate your SSL certificate with a distribution that is enabled. For more information, see http://aws.amazon.com/cloudfront/pricing.
  - **Only Clients that Support Server Name Indication (SNI)**: Older browsers or other clients that don't support SNI must use another method to access your content.

  For more information, see Choosing How CloudFront Serves HTTPS Requests (p. 171).

f. Click **Yes, Edit**.

3. In the CloudFront console, on the **General** tab for your distribution, confirm that the status of your distribution has changed to **Deployed**. If you try to use an alternate domain name before the updates to your distribution have been deployed, the links you create in the following steps might not work.
4. Using the method provided by your DNS service provider, add a CNAME resource record set to the hosted zone for your domain. This new CNAME resource record set will redirect DNS queries from your domain (for example, www.example.com) to the CloudFront domain name for your distribution (for example, d111111abcdef8.cloudfront.net). For more information, see the documentation provided by your DNS service provider.

   If you're using Amazon Route 53 as your DNS service, you can create an alias resource record set instead of a CNAME. With an alias resource record set, you don't pay for Amazon Route 53 queries. In addition, you can create an alias resource record set for a domain name at the zone apex (example.com), which DNS doesn't allow for CNAMEs. For more information, go to Routing Queries to an Amazon CloudFront Distribution in the *Amazon Route 53 Developer Guide*.

   Important
   If you already have an existing CNAME record for your domain name, update that resource record set or replace it with a new one that points to the CloudFront domain name for your distribution.
   In addition, confirm that your CNAME resource record set points to your distribution's domain name and not to one of your origin servers.

5. Using dig or a similar tool, confirm that the CNAME resource record set that you created in Step 4 points to the domain name for your distribution. For more information about dig, go to http://www.kloth.net/services/dig.php.

   The following example shows a dig request on the images.example.com domain, as well as the relevant part of the response.

   ```
   [prompt]> dig images.example.com

   ; <<> DiG 9.3.3rc2 <<> images.example.com
   ;; global options:  printcmd
   ;; Got answer:
   ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15917
   ;; flags: qr rd ra; QUERY: 1, ANSWER: 9, AUTHORITY: 2, ADDITIONAL: 0

   ;; QUESTION SECTION:
   ;images.example.com.     IN    A

   ;; ANSWER SECTION:
   images.example.com. 10800 IN  CNAME  d111111abcdef8.cloudfront.net.
   ...
   ...
   ```

   The line in the Answer Section shows a CNAME resource record set that routes queries for images.example.com to the CloudFront distribution domain name d111111abcdef8.cloudfront.net. The CNAME resource record set is configured correctly if the name on the right side of `CNAME` is the domain name for your CloudFront distribution. If that is any other value, for example, the domain name for your Amazon S3 bucket, then the CNAME resource record set is configured incorrectly. In that case, go back to Step 4 and correct the CNAME record to point to the domain name for your distribution.
6. Test the alternate domain name by creating some test links that use your domain name in the URL instead of the CloudFront domain name for your distribution.
7. In your application, change the links for your objects to use your alternate domain name instead of the domain name of your CloudFront distribution.

Choosing the Price Class for a CloudFront Distribution

CloudFront has edge locations all over the world. Our cost for each edge location varies and, as a result, the price that we charge you varies depending on the edge location from which CloudFront serves your requests.

CloudFront edge locations are grouped into geographic regions, and we've grouped regions into price classes. The default price class includes all regions. Another price class includes most regions (the United States; Europe; Hong Kong, Korea, and Singapore; Japan; and India regions) but excludes the most-expensive regions. A third price class includes only the least-expensive regions (the United States and Europe regions).

By default, CloudFront responds to requests for your objects based only on performance: objects are served from the edge location for which latency is lowest for that viewer. If you're willing to accept higher latency for your viewers in some geographic regions in return for lower cost, you can choose a price class that doesn't include all CloudFront regions. Although CloudFront will serve your objects only from the edge locations in that price class, it still serves content from the edge location that has the lowest latency among the edge locations in your selected price class. However, some of your viewers, especially those in geographic regions that are not in your price class, may see higher latency than if your content were being served from all CloudFront edge locations. For example, if you choose the price class that includes only the United States and Europe, viewers in Australia and in Asia may experience higher latency than if you choose the price class that includes Australia and Asia.

If you choose a price class that does not include all edge locations, CloudFront may still occasionally serve requests for your content from an edge location in a region that is not included in your price class. When this happens, you are not charged the rate for the more expensive region from which your objects were served. Instead, you're charged the rate for the least-expensive region in your selected price class.

You can choose a price class when you create a CloudFront distribution, or you can update an existing distribution using the CloudFront console or the CloudFront API. To find the applicable topic about creating or updating a web or an RTMP distribution using the CloudFront console or API, see Working with Distributions (p. 25).

For more information about CloudFront pricing and price classes, go to Amazon CloudFront Pricing.

Using CloudFront with Amazon S3

If you currently distribute content from your Amazon S3 bucket using a CNAME, you can migrate to CloudFront with no disruption by using the following process.

If your objects are stored in Amazon S3, you can either have your users get your objects directly from your Amazon S3 bucket, or you can configure CloudFront to get your objects from Amazon S3 and have your users get your objects from CloudFront.

If your users access your objects frequently, you can lower the cost of delivering your objects by adding CloudFront because, at higher usage, the price for CloudFront data transfer is lower than the price for Amazon S3 data transfer. In addition, downloads are faster with CloudFront than with Amazon S3 alone because your objects are stored closer to your users.

> **Note**
> If you want CloudFront to respect Amazon S3 cross-origin resource sharing settings, configure CloudFront to forward the `Origin` header to Amazon S3. For more information, see Configuring CloudFront to Cache Objects Based on Request Headers (p. 78).

If you currently distribute content directly from your Amazon S3 bucket using your own domain name (such as example.com) instead of the domain name of your Amazon S3 bucket (such as MyAWSBucket.s3.amazonaws.com), you can add CloudFront with no disruption by using the following process.

Process for Adding CloudFront When You're Already Distributing Your Content from Amazon S3

1. Create a CloudFront distribution using the process described in the applicable topic:

   - Task List for Creating a Web Distribution (p. 36)
   - Task List for Streaming Media Files Using RTMP (p. 62)

   When you create the distribution, specify the name of your Amazon S3 bucket as the origin server.

   > **Important**
   > For your bucket to work with CloudFront, the name must conform to DNS naming requirements. For more information, go to Bucket Restrictions and Limitations in the *Amazon Simple Storage Service Developer Guide*.

   If you're using a CNAME with Amazon S3, specify the CNAME for your distribution, too.

2. Create a test web page that contains links to publicly readable objects in your Amazon S3 bucket, and test the links. For this initial test, use the CloudFront domain name of your distribution in the object URLs, for example, `http://d111111abcdef8.cloudfront.net/images/image.jpg`.

   For more information about format of CloudFront URLs, see Format of URLs for CloudFront Objects (p. 72).

3. If you're using Amazon S3 CNAMEs, your application uses your domain name (for example, example.com) to reference the objects in your Amazon S3 bucket instead of using the name of your bucket (for example, myawsbucket.s3.amazonaws.com). To continue using your domain name to reference objects instead of using the CloudFront domain name for your distribution (for example, d111111abcdef8.cloudfront.net), you need to update your settings with your DNS service provider.

   For Amazon S3 CNAMEs to work, your DNS service provider has a CNAME resource record set for your domain that currently routes end-user queries for the domain to your Amazon S3 bucket. When someone requests the object:

   ```
   http://example.com/images/image.jpg
   ```

   the request is automatically rerouted, and the object they see is:

   ```
   http://myawsbucket.s3.amazonaws.com/images/image.jpg
   ```

   To route queries to your CloudFront distribution instead of your Amazon S3 bucket, you need to use the method provided by your DNS service provider to update the CNAME resource record set for your domain. This updated CNAME record will start to redirect DNS queries from your domain to the CloudFront domain name for your distribution. For more information, see the documentation provided by your DNS service provider.

   Note
   If you're using Amazon Route 53 as your DNS service, you can use either a CNAME resource record set or an alias resource record set. For information about editing resource record sets, see Editing Resource Record Sets. For information about alias resource record sets, see Choosing Between Alias and Non-Alias Resource Record Sets. Both topics are in the *Amazon Route 53 Developer Guide*.

   For more information about using CNAMEs with CloudFront, see Using Alternate Domain Names (CNAMEs) (p. 29).

   After you update the CNAME resource record set, it can take up to 72 hours for the change to propagate throughout the DNS system, although it usually happens faster. During this time, some requests for your content will continue to be routed to your Amazon S3 bucket, and others will be routed to CloudFront.

Changes to the CloudFront API

Beginning with the 2012-05-05 version of the CloudFront API, we made substantial changes to the format of the XML document that you include in the request body when you create or update a web distribution or an RTMP distribution, and when you invalidate objects. With previous versions of the API, we discovered that it was too easy to accidentally delete one or more values for an element that accepts multiple values, for example, CNAMEs and trusted signers. Our changes for the 2012-05-05 release are intended to prevent these accidental deletions and to notify you when there's a mismatch between the number of values you say you're specifying in the `Quantity` element and the number of values you're actually specifying.

Note the following about using the 2012-05-05 API version or later with web and RTMP distributions that were created using earlier API versions:

- You cannot use versions of the API earlier than 2012-05-05 to update a web distribution that was created or updated using the 2012-05-05 or later CloudFront API.
- You can use the new API version to get a list of distributions, get information about a distribution, or get distribution configuration. CloudFront returns an XML document in the new XML format.
- To update a distribution that was created using an earlier API version, use the 2012-05-05 or later version of GET Distribution or GET Streaming Distribution to get an XML document in the new XML format, change the data as applicable, and use the 2012-05-05 or later version of PUT Distribution Config or PUT Streaming Distribution Config to submit the changes to CloudFront.
- You can use the new API to delete a distribution that was created using an earlier API version. The distribution must already be disabled.

Working with Web Distributions

Topics

- Task List for Creating a Web Distribution (p. 36)
- Creating a Web Distribution Using the CloudFront Console (p. 37)
- Testing Your Web Distribution (p. 38)
- Using Amazon S3 Origins and Custom Origins for Web Distributions (p. 39)
- Values that You Specify When You Create or Update a Web Distribution (p. 40)
- Values that CloudFront Displays in the Console When You Create or Update a Web Distribution (p. 55)
- Requirements and Recommendations for Using Amazon EC2 and Other Custom Origins (p. 56)
- Restricting the Geographic Distribution of Your Content (p. 56)
- Configuring On-Demand Smooth Streaming (p. 57)
- Configuring On-Demand Progressive Downloads (p. 58)
- Configuring On-Demand Apple HTTP Live Streaming (HLS) (p. 59)

This section describes how you configure and manage CloudFront web distributions. For a basic explanation of distributions, see Working with Distributions (p. 25). For information about CloudFront RTMP distributions, see Working with RTMP Distributions (p. 60).

Task List for Creating a Web Distribution

The following task list summarizes the process for creating a web distribution.

To Create a Web Distribution

1. Create one or more Amazon S3 buckets or configure HTTP servers as your origin servers. An origin is the location where you store the original version of your web content. When CloudFront gets a request for your files, it goes to the origin to get the files that it distributes at edge locations. You can use any combination of up to 10 Amazon S3 buckets and HTTP servers as your origin servers.

   If you're using Amazon S3, note that the name of your bucket must be all lowercase and cannot contain spaces.

   If you're using an Amazon EC2 server or another custom origin, review Requirements and Recommendations for Using Amazon EC2 and Other Custom Origins (p. 56).

2. Upload your content to your origin servers. If you don't want to restrict access to your content using CloudFront signed URLs, make the objects publicly readable.

   Caution
   You are responsible for ensuring the security of your origin server. You must ensure that CloudFront has permission to access the server and that the security settings are appropriate to safeguard your content.

3. Create your CloudFront web distribution:

   - For more information about creating a web distribution using the CloudFront console, see Creating a Web Distribution Using the CloudFront Console (p. 37).
   - For information about creating a web distribution using the CloudFront API, go to POST Distribution in the *Amazon CloudFront API Reference*.

4. Optional: If you created your distribution using the CloudFront console, create more cache behaviors or origins for your distribution. For more information, see To List, View, and Update CloudFront Distributions Using the CloudFront Console (p. 27).
5. Test your web distribution. For more information, see Testing Your Web Distribution (p. 38).
6. Develop your website or application to access your content using the domain name that CloudFront returned after you created your distribution in Step 3. For example, if CloudFront returns d111111abcdef8.cloudfront.net as the domain name for your distribution, the URL for the file `image.jpg` in an Amazon S3 bucket or in the root directory on an HTTP server will be `http://d111111abcdef8.cloudfront.net/image.jpg`.

   If you specified one or more alternate domain names (CNAMEs) when you created your distribution, you can use your own domain name. In that case, the URL for `image.jpg` might be `http://www.example.com/image.jpg`.

   Note the following:

   - If you want to use signed URLs to restrict access to your content, see Serving Private Content through CloudFront (p. 118).
   - If you want to serve compressed content, see Serving Compressed Files (p. 98).
   - For information about CloudFront request and response behavior for Amazon S3 and custom origins, see Request and Response Behavior (p. 101).

Creating a Web Distribution Using the CloudFront Console

The following procedure explains how to create a web distribution using the CloudFront console. If you want to create a web distribution using the CloudFront API, go to POST Distribution in the *Amazon CloudFront API Reference*.

For the current limit on the number of web distributions that you can create for each AWS account, see Amazon CloudFront Limits in the *Amazon Web Services General Reference*. To request a higher limit, go to https://aws.amazon.com/support/createCase?type=service_limit_increase&serviceLimitIncreaseType=cloudfront-distributions.

To create a CloudFront web distribution using the CloudFront console

1. Sign in to the AWS Management Console and open the Amazon CloudFront console at https://console.aws.amazon.com/cloudfront/.
2. Click **Create Distribution**.
3. On the first page of the **Create Distribution Wizard**, accept the default selection, **Web**, and click **Continue**.
4. Specify settings for the distribution. For more information, see Values that You Specify When You Create or Update a Web Distribution (p. 40).
5. Click **Create Distribution**.
6. After CloudFront creates your distribution, the value of the **Status** column for your distribution will change from **InProgress** to **Deployed**. If you chose to enable the distribution, it will then be ready to process requests. This should take less than 15 minutes.

   The domain name that CloudFront assigns to your distribution appears in the list of distributions. (It also appears on the **General** tab for a selected distribution.)

When your distribution is deployed, confirm that you can access your content using your new CloudFront URL or CNAME. For more information, see Testing Your Web Distribution (p. 38).

Testing Your Web Distribution

After you've created your distribution, CloudFront knows where your origin server is, and you know the domain name associated with the distribution. You can create links to your objects using the CloudFront domain name, and CloudFront will serve the objects to your web page or application.

> **Note**
> You must wait until the status of the distribution changes to **Deployed** before you can test your links.

To create links to objects in a web distribution

1. Copy the following HTML code into a new file, replace *`domain-name`* with your distribution's domain name, and replace *`object-name`* with the name of your object.

   ```
   <html>
   <head>My CloudFront Test</head>
   <body>
   <p>My text content goes here.</p>
   <p><img src="http://domain-name/object-name" alt="my test image"
   </body>
   </html>
   ```

   For example, if your domain name were `d111111abcdef8.cloudfront.net` and your object were `image.jpg`, the URL for the link would be:

   `http://d111111abcdef8.cloudfront.net/image.jpg`.

   If your object is in a folder on your origin server, then the folder must also be included in the URL. For example, if image.jpg were located in the images folder on your origin server, then the URL would be:

   `http://d111111abcdef8.cloudfront.net/images/image.jpg`
2. Save the HTML code in a file that has a .html filename extension.

3. Open your web page in a browser to ensure that you can see your object.

The browser returns your page with the embedded image file, served from the edge location that CloudFront determined was appropriate to serve the object.

Using Amazon S3 Origins and Custom Origins for Web Distributions

When you create a web distribution, you specify where CloudFront sends requests for the files that it distributes to edge locations. CloudFront supports using Amazon S3 buckets and HTTP servers (for example, web servers) as origins.

Using Amazon S3 Buckets for Your Origin

When you use Amazon S3 as an origin for your distribution, you place any objects that you want CloudFront to deliver in an Amazon S3 bucket. You can use any method that is supported by Amazon S3 to get your objects into Amazon S3, for example, the Amazon S3 console or API, or a third-party tool. You can create a hierarchy in your bucket to store the objects, just as you would with any other Amazon S3 bucket.

Using an existing Amazon S3 bucket as your CloudFront origin server doesn't change the bucket in any way; you can still use it as you normally would to store and access Amazon S3 objects at the standard Amazon S3 price. You incur regular Amazon S3 charges for storing the objects in the bucket. For more information about the charges to use CloudFront, see CloudFront Billing and Usage Reports (p. 7).

> **Important**
> For your bucket to work with CloudFront, the name must conform to DNS naming requirements. For more information, go to Bucket Restrictions and Limitations in the *Amazon Simple Storage Service Developer Guide*.

When you specify the Amazon S3 bucket that you want CloudFront to get objects from, how you specify the bucket name depends on whether you have configured the bucket as a website endpoint:

The bucket is not configured as a website endpoint

In general, use the following format:

```
bucket-name.s3.amazonaws.com
```

If your bucket is in the US Standard region and you want Amazon S3 to route requests to a facility in Northern Virginia, use the following format:

```
bucket-name.s3-external-1.amazonaws.com
```

When you specify the bucket name in this format, you can use the following CloudFront features:

- Configure CloudFront to communicate with your Amazon S3 bucket using SSL. For more information, see Using an HTTPS Connection to Access Your Objects (p. 168).
- Use an origin access identity to require that your users access your content using CloudFront URLs, not by using Amazon S3 URLs. For more information, see Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content (p. 123).
- Update the content of your bucket by submitting `POST` and `PUT` requests to CloudFront. For more information, see HTTP Methods (p. 103) in the topic How CloudFront Processes and Forwards Requests to Your Amazon S3 Origin Server (p. 101).

The bucket is configured as a website endpoint

Enter the Amazon S3 static website hosting endpoint for your bucket. This value appears in the Amazon S3 console, on the **Properties** page under **Static Website Hosting**.

When you specify the bucket name in this format, you can use Amazon S3 redirects and Amazon S3 custom error documents. (CloudFront also provides custom error pages. For more information, see Customizing Error Responses (p. 92).) For more information about Amazon S3 features, see the Amazon S3 documentation.

Do not specify the bucket using the following formats:

- The Amazon S3 path style, `s3.amazonaws.com/bucket-name`
- The Amazon S3 CNAME, if any

Using Amazon EC2 or Other Custom Origins

A custom origin is an HTTP server, for example, a web server. The HTTP server can be an Amazon EC2 instance or an HTTP server that you manage privately. When you use a custom origin, you specify the DNS name of the server, along with the HTTP and HTTPS ports and the protocol that you want CloudFront to use when fetching objects from your origin.

Most CloudFront features are supported when you use a custom origin with the following exceptions:

- **RTMP distributions**—Not supported.
- **Private content**—Although you can use a signed URL to distribute content from a custom origin, for CloudFront to access the custom origin, the origin must remain publicly accessible. For more information, see Serving Private Content through CloudFront (p. 118).

For information about requirements and recommendations when using custom origins, see Requirements and Recommendations for Using Amazon EC2 and Other Custom Origins (p. 56).

Values that You Specify When You Create or Update a Web Distribution

When you create a new web distribution or update an existing distribution, you specify the following values. For information about creating or updating a web distribution using the CloudFront console, see the applicable topic:

- Working with Web Distributions (p. 36)
- Listing, Viewing, and Updating CloudFront Distributions (p. 27)

Delivery Method (p. 42)

Origin Settings (p. 42)

- Origin Domain Name (p. 42)
- Origin ID (p. 43)
- Restrict Bucket Access (Amazon S3 Only) (p. 43)
- Origin Access Identity (Amazon S3 Only) (p. 43)
- Comment for New Identity (Amazon S3 Only) (p. 43)
- Your Identities (Amazon S3 Only) (p. 43)
- Grant Read Permissions on Bucket (Amazon S3 Only) (p. 43)
- HTTP Port (Amazon EC2 and Other Custom Origins Only) (p. 44)

- HTTPS Port (Amazon EC2 and Other Custom Origins Only) (p. 44)
- Origin Protocol Policy (Amazon EC2, Amazon S3 Buckets Configured as Website Endpoints, and Other Custom Origins Only) (p. 44)

Cache Behavior Settings (p. 44)

- Path Pattern (p. 45)
- Origin (Existing Distributions Only) (p. 47)
- Viewer Protocol Policy (p. 47)
- Allowed HTTP Methods (p. 47)
- Forward Headers (p. 47)
- Whitelist Headers (p. 48)
- Object Caching (p. 48)
- Minimum TTL (p. 48)
- Forward Cookies (Amazon EC2 and Other Custom Origins Only) (p. 48)
- Whitelist Cookies (Amazon EC2 and Other Custom Origins Only) (p. 49)
- Forward Query Strings (p. 49)
- Smooth Streaming (p. 49)
- Restrict Viewer Access (Use Signed URLs) (p. 49)
- Trusted Signers (p. 49)
- AWS Account Numbers (p. 50)

Distribution Details (p. 50)

- Price Class (p. 50)
- Alternate Domain Names (CNAMEs) (p. 50)
- SSL Certificate (p. 51)
- Clients Supported (p. 51)
- Default Root Object (p. 52)
- Logging (p. 52)
- Bucket for Logs (p. 52)
- Log Prefix (p. 52)
- Cookie Logging (p. 52)
- Comment (p. 53)
- Distribution State (p. 53)

Custom Error Pages and Error Caching (p. 53)

- Error Code (p. 53)
- Response Page Path (p. 53)
- Response Code (p. 54)
- Error Caching Minimum TTL (p. 54)

Restrictions (p. 54)

- Enable Geo Restriction (p. 54)
- Restriction Type (p. 54)
- Countries (p. 54)

Delivery Method

You specify the delivery method when you create a distribution. For a web distribution, this value is always **Web**. You can't change the delivery method for an existing distribution.

Origin Settings

When you create or update a distribution, you provide information about one or more locations—known as origins—where you store the original versions of your web content. CloudFront gets your web content from your origins and serves it to viewers via a world-wide network of edge servers. Each origin is either an Amazon S3 bucket or an HTTP server, for example, a web server.

For the current limit on the number of origins that you can create for a distribution, see Amazon CloudFront Limits in the *Amazon Web Services General Reference*. To request a higher limit, go to https://aws.amazon.com/support/createCase?type=service_limit_increase&serviceLimitIncreaseType=cloudfront-distributions.

If you want to delete an origin, you must first edit or delete the cache behaviors that are associated with that origin.

> **Caution**
> If you delete an origin, confirm that files that were previously served by that origin are available in another origin and that your cache behaviors are now routing requests for those files to the new origin.

When you create or update a distribution, you specify the following values for each origin.

Origin Domain Name

The DNS domain name of the Amazon S3 bucket or HTTP server from which you want CloudFront to get objects for this origin, for example, `myawsbucket.s3.amazonaws.com` or `www.example.com`.

If your origin is an HTTP server, type the domain name of the resource. The files must be publicly readable.

If your origin is an Amazon S3 bucket, in the CloudFront console, click in the **Origin Domain Name** field, and a list enumerates the Amazon S3 buckets that are associated with the current AWS account. Note the following:

- If the bucket is configured as a website, enter the Amazon S3 static website hosting endpoint for your bucket; do not select the bucket name from the list in the **Origin Domain Name** field. The static website hosting endpoint appears in the Amazon S3 console, on the **Properties** page under **Static Website Hosting**.
- If you're using a bucket from a different AWS account and if the bucket is not configured as a website, type the name in the following format:

  ```
  bucket-name.s3.amazonaws.com
  ```

  If your bucket is in the US Standard region and you want Amazon S3 to route requests to a facility in Northern Virginia, use the following format:

  ```
  bucket-name.s3-external-1.amazonaws.com
  ```
- The files must be publicly readable unless you secure your content in Amazon S3 by using a CloudFront origin access identity. For more information, see Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content (p. 123).

Important
If the origin is an Amazon S3 bucket, the bucket name must conform to DNS naming requirements. For more information, go to Bucket Restrictions and Limitations in the *Amazon Simple Storage Service Developer Guide*.

When you change the value of **Origin Domain Name** for an origin, CloudFront immediately begins replicating the change to CloudFront edge locations. Until the distribution configuration is updated in a given edge location, CloudFront will continue to forward requests to the previous HTTP server or Amazon S3 bucket. As soon as the distribution configuration is updated in that edge location, CloudFront begins to forward requests to the new HTTP server or Amazon S3 bucket.

Changing the origin does not require CloudFront to repopulate edge caches with objects from the new origin. As long as the viewer requests in your application have not changed, CloudFront will continue to serve objects that are already in an edge cache until the TTL on each object expires or until seldom-requested objects are evicted.

Origin ID

A string that uniquely distinguishes this origin from other origins in this distribution. If you create cache behaviors in addition to the default cache behavior, you use the origin ID that you specify here to identify the origin to which you want CloudFront to route a request when the request matches the path pattern for that cache behavior. For more information, see Cache Behavior Settings (p. 44).

Restrict Bucket Access (Amazon S3 Only)

Click **Yes** if you want to require end users to access objects in an Amazon S3 bucket by using only CloudFront URLs, not by using Amazon S3 URLs. Then specify the applicable values.

Click **No** if you want end users to be able to access objects using either CloudFront URLs or Amazon S3 URLs.

For more information, see Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content (p. 123).

Origin Access Identity (Amazon S3 Only)

If you chose **Yes** for **Restrict Bucket Access**, choose whether to create a new origin access identity or use an existing one that is associated with your AWS account. If you already have an origin access identity, we recommend that you reuse it to simplify maintenance. For more information about origin access identities, see Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content (p. 123).

Comment for New Identity (Amazon S3 Only)

If you chose **Create a New Identity** for **Origin Access Identity**, enter a comment that identifies the new origin access identity. CloudFront will create the origin access identity when you create this distribution.

Your Identities (Amazon S3 Only)

If you chose **Use an Existing Identity** for **Origin Access Identity**, choose the origin access identity that you want to use. You cannot use an origin access identity that is associated with another AWS account.

Grant Read Permissions on Bucket (Amazon S3 Only)

If you want CloudFront to automatically grant the origin access identity the permission to read objects in your Amazon S3 bucket, click **Yes, Update Bucket Policy**.

Important
If you click **Yes, Update Bucket Policy**, CloudFront updates the bucket policy to grant the specified origin access identity the permission to read objects in your bucket. However, CloudFront does not remove existing permissions in the bucket policy or permissions on individual objects. If users currently have permission to access the objects in your bucket using Amazon S3 URLs, they will still have that permission after CloudFront updates your bucket policy. To view or change the existing bucket policy and the existing permissions on the objects in your bucket, use a method provided by Amazon S3. For more information, see Granting the Origin Access Identity Permission to Read Objects in Your Amazon S3 Bucket (p. 125).

If you want to update permissions manually, for example, if you want to update ACLs on your objects instead of updating bucket permissions, click **No, I will Update Permissions**.

Origin Protocol Policy (Amazon EC2, Amazon S3 Buckets Configured as Website Endpoints, and Other Custom Origins Only)

The protocol policy that you want CloudFront to use when fetching objects from your origin server. Choose the applicable value:

- **HTTP Only:** CloudFront only uses HTTP to access the origin.

  Important
  If your Amazon S3 bucket is configured as a website endpoint, you must specify **HTTP Only**. Amazon S3 doesn't support HTTPS connections in that configuration.

- **Match Viewer:** CloudFront fetches objects from your origin using HTTP or HTTPS, depending on the protocol of the viewer request. CloudFront caches the object only once even if viewers make requests using both HTTP and HTTPS protocols.

  Important
  For HTTPS viewer requests that CloudFront forwards to this origin, one of the domain names in the SSL certificate on your origin server must match the domain name that you specify for **Origin Domain Name**. Otherwise, CloudFront responds to the viewer requests with an HTTP status code 502 (bad gateway) instead of the requested object. For more information, see How to Require HTTPS for Communication between Viewers, CloudFront, and Your Origin (p. 169).

HTTP Port (Amazon EC2 and Other Custom Origins Only)

Optional. The HTTP port that the custom origin listens on. Valid values include ports 80, 443, and 1024 to 65535. The default value is port 80.

HTTPS Port (Amazon EC2 and Other Custom Origins Only)

Optional. The HTTPS port that the custom origin listens on. Valid values include ports 80, 443, and 1024 to 65535. The default value is port 443.

Cache Behavior Settings

A cache behavior lets you configure a variety of CloudFront functionality for a given URL path pattern for files on your website. For example, one cache behavior might apply to all `.jpg` files in the `images` directory on a web server that you're using as an origin server for CloudFront. The functionality you can configure for each cache behavior includes:

- The path pattern.

- If you have configured multiple origins for your CloudFront distribution, which origin you want CloudFront to forward your requests to.
- Whether to forward query strings to your origin.
- Whether accessing the specified files requires signed URLs.
- Whether to require end users to use HTTPS to access those files.
- The minimum amount of time that those files stay in the CloudFront cache regardless of the value of any `Cache-Control` headers that your origin adds to the files.

When you create a new distribution, you specify settings for the default cache behavior, which automatically forwards all requests to the origin that you specify when you create the distribution. After you create a distribution, you can create additional cache behaviors that define how CloudFront responds when it receives a request for objects that match a path pattern, for example, `*.jpg`. If you create additional cache behaviors, the default cache behavior is always the last to be processed. Other cache behaviors are processed in the order in which they're listed in the CloudFront console or, if you're using the CloudFront API, the order in which they're listed in the `DistributionConfig` element for the distribution. For more information, see Path Pattern (p. 45).

When you create a cache behavior, you specify the one origin from which you want CloudFront to get objects. As a result, if you want CloudFront to distribute objects from all of your origins, you must have at least as many cache behaviors (including the default cache behavior) as you have origins. For example, if you have two origins and only the default cache behavior, the default cache behavior will cause CloudFront to get objects from one of the origins, but the other origin will never be used.

For the current limit on the number of cache behaviors that you can add to a distribution, see Amazon CloudFront Limits in the *Amazon Web Services General Reference*. To request a higher limit, go to https://aws.amazon.com/support/createCase?type=service_limit_increase&serviceLimitIncreaseType=cloudfront-distributions.

Path Pattern

A path pattern (for example, `/images/*.jpg`) specifies which requests you want this cache behavior to apply to. When CloudFront receives an end-user request, the requested path is compared with path patterns in the order in which cache behaviors are listed in the distribution. The first match determines which cache behavior is applied to that request. For example, suppose you have three cache behaviors with the following three path patterns, in this order:

- `/images/*.jpg`
- `/images/*`
- `/*.gif`

A request for the file `/images/sample.gif` doesn't satisfy the first path pattern, so the associated cache behaviors are not be applied to the request. The file does satisfy the second path pattern, so the cache behaviors associated with the second path pattern are applied even though the request also matches the third path pattern.

Note

When you create a new distribution, the value of **Path Pattern** for the default cache behavior is set to ***** (all files) and cannot be changed. This value causes CloudFront to forward all requests for your objects to the origin that you specified in the Origin Domain Name (p. 42) field. If the request for an object does not match the path pattern for any of the other cache behaviors, CloudFront applies the behavior that you specify in the default cache behavior.

Caution

Define path patterns and their sequence carefully or you may give end users undesired access to your content. For example, suppose a request matches the path pattern for two cache behaviors. The first cache behavior does not require signed URLs and the second cache behavior does require signed URLs. End users will be able to access the objects without using a signed URL because CloudFront processes the cache behavior associated with the first match.

The path you specify applies to requests for all files in the specified directory and in subdirectories below the specified directory. CloudFront does not consider query strings or cookies when evaluating the path pattern. For example, if an /images directory contains product1 and product2 subdirectories, the path pattern /images/*.jpg applies to requests for any .jpg file in the /images, /images/product1, and /images/product2 directories. If you want to apply a different cache behavior to the files in the /images/product1 directory than the files in the /images and /images/product2 directories, create a separate cache behavior for /images/product1 and move that cache behavior to a position above (before) the cache behavior for the /images directory.

You can use the following wildcard characters in your path pattern:

- * matches 0 or more characters.
- ? matches exactly 1 character.

The following examples show how the wildcard characters work:

Path pattern	Files that match the path pattern
/*.jpg	All .jpg files
/images/*.jpg	All .jpg files in the /images directory and in subdirectories under the /images directory
/a*.jpg	• All .jpg files for which the filename begins with a, for example, apple.jpg and appalachian_trail_2012_05_21.jpg • All .jpg files for which the file path begins with a, for example, /abra/cadabra/magic.jpg.
/a??.jpg	All .jpg files for which the filename begins with a and is followed by exactly two other characters, for example, ant.jpg and abe.jpg
/*.doc*	All files for which the filename extension begins with .doc, for example, .doc, .docx, and .docm files. You can't use the path pattern *.doc? in this case, because that path pattern wouldn't apply to requests for .doc files; the ? wildcard character replaces exactly one character.

The maximum length of a path pattern is 255 characters. The value can contain any of the following characters:

- A-Z, a-z

  Path patterns are case sensitive, so the path pattern /*.jpg doesn't apply to the file /LOGO.JPG.
- 0-9
- _ - . * $ / ~ " ' @ : +
- &, passed and returned as &amp;

Origin (Existing Distributions Only)

Enter the value of **Origin ID** for an existing origin. This identifies the origin that you want CloudFront to route requests to when a request (such as http://example.com/logo.jpg) matches the path pattern for a cache behavior (such as *.jpg) or for the default cache behavior (*).

Viewer Protocol Policy

Choose the protocol policy that you want viewers to use to access your content in CloudFront edge locations:

- **HTTP and HTTPS**: Viewers can use both protocols.
- **Redirect HTTP to HTTPS**: Viewers can use both protocols, but HTTP requests are automatically redirected to HTTPS requests.
- **HTTPS Only**: Viewers can only access your content if they're using HTTPS.

For more information, see Using an HTTPS Connection to Access Your Objects (p. 168).

Allowed HTTP Methods

Specify which HTTP methods you want CloudFront to process and forward to your origin:

- **GET, HEAD:** You can only use CloudFront to get objects from your origin or to get object headers.
- **GET, HEAD, PUT, POST, PATCH, DELETE, OPTIONS:** You can use CloudFront to get, add, update, and delete objects, and to get object headers. In addition, you can perform other POST operations such as submitting data from a web form.

  Note
  CloudFront caches responses to `GET` and `HEAD` requests. CloudFront does not cache responses to requests that use the other methods.

Caution
If you choose **GET, HEAD, PUT, POST, PATCH, DELETE, OPTIONS**, you may need to restrict access to your Amazon S3 bucket or to your custom origin so users can't perform operations that you don't want them to:

- **If you're using Amazon S3 as an origin for your distribution:** Create a CloudFront origin access identity to restrict access to your Amazon S3 content, and grant the origin access identity the applicable permissions. For example, if you configure CloudFront to accept and forward these methods *only* because you want to use `PUT`, you must still configure Amazon S3 bucket policies or ACLs to handle `DELETE` requests appropriately. For more information, see Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content (p. 123).
- **If you're using a custom origin:** Configure your origin server to handle all methods. For example, if you configure CloudFront to accept and forward these methods *only* because you want to use `POST`, you must still configure your origin server to handle `DELETE` requests appropriately.

Forward Headers

Specify whether you want CloudFront to forward request headers to your origin server and to cache objects based on header values:

- **All** – CloudFront forwards all headers to your origin for a cache behavior.

> **Important**
> If you configure CloudFront to forward all headers to your origin, CloudFront doesn't cache the objects associated with this cache behavior. Instead, it sends every request to the origin.

- **Whitelist** – CloudFront forwards only the specified headers to the origin. Use **Whitelist Headers** to choose the headers that you want CloudFront to forward.
- **None (Improves Caching)** – CloudFront forwards default headers to the origin, but it doesn't cache your objects based on the header values.

If you're using an Amazon S3 bucket as your origin, note that Amazon S3 processes only the `Origin` header. It ignores all other headers in a request.

For more information about forwarding headers to the origin, see Configuring CloudFront to Cache Objects Based on Request Headers (p. 78). For a list of HTTP headers and information about whether CloudFront forwards the header to the origin by default, see HTTP Request Headers and CloudFront Behavior (p. 109).

Whitelist Headers

Specify the headers that you want CloudFront to consider when caching your objects. Select headers from the list of available headers and click **Add**. To forward a custom header, enter the name of the header in the field, and click **Add Custom**.

For the current limit on the number of headers that you can whitelist for each cache behavior, see Amazon CloudFront Limits in the *Amazon Web Services General Reference*. To request a higher limit, go to https://aws.amazon.com/support/createCase?type=service_limit_increase&serviceLimitIncreaseType=cloudfront-distributions.

Object Caching

If your origin server is adding a `Cache-Control` header to your objects to control how long the objects stay in the CloudFront cache, choose **Use Origin Cache Headers**.

To specify a minimum time that your objects stay in the CloudFront cache regardless of `Cache-Control` headers, choose **Customize**. Then, in the **Minimum TTL** field, specify the minimum number of seconds that you want objects to stay in the CloudFront cache even if `Cache-Control` headers specify a lower value.

Minimum TTL

Specify the minimum amount of time, in seconds, that you want objects to stay in CloudFront caches before CloudFront queries your origin to see whether the object has been updated.

> **Important**
> If you configure CloudFront to forward all headers to your origin for a cache behavior, CloudFront never caches the associated objects. Instead, CloudFront forwards all requests for those objects to the origin. The value of **Minimum TTL** must be 0.

For more information, see Specifying How Long Objects Stay in a CloudFront Edge Cache (Expiration) (p. 83).

Forward Cookies (Amazon EC2 and Other Custom Origins Only)

Specify whether you want CloudFront to forward cookies to your origin server and, if so, which ones. If you choose to forward only selected cookies (a whitelist of cookies), enter the cookie names in the

Whitelist Cookies field. If you choose **All**, CloudFront forwards all cookies regardless of how many your application uses.

Amazon S3 doesn't process cookies, and forwarding cookies to the origin reduces cacheability. For cache behaviors that are forwarding requests to an Amazon S3 origin, choose **None** for **Forward Cookies**.

For more information about forwarding cookies to the origin, go to Configuring CloudFront to Cache Objects Based on Cookies (p. 76).

Whitelist Cookies (Amazon EC2 and Other Custom Origins Only)

If you chose **Whitelist** in the **Forward Cookies** list, in the **Whitelist Cookies** field, enter the names of cookies that you want CloudFront to forward to your origin server for this cache behavior. Enter each cookie name on a new line.

For the current limit on the number of cookie names that you can whitelist for each cache behavior, see Amazon CloudFront Limits in the *Amazon Web Services General Reference*. To request a higher limit, go to https://aws.amazon.com/support/createCase?type=service_limit_increase&serviceLimitIncreaseType=cloudfront-distributions.

Forward Query Strings

If your origin server returns different versions of an object based on a query string in the URL, click **Yes**. If your origin returns the same version of an object regardless of the query string, click **No**. This increases the likelihood that CloudFront can serve a request from the cache, which improves performance and reduces the load on your origin. For more information about query strings, see Configuring CloudFront to Cache Based on Query String Parameters (p. 75).

Smooth Streaming

Click **Yes** if you want to distribute media files in the Microsoft Smooth Streaming format using the origin that is associated with this cache behavior. Otherwise, click **No**.

> **Note**
> If you specify **Yes**, you can still distribute other content using this cache behavior if the content matches the value of **Path Pattern**.

For more information, see Configuring On-Demand Smooth Streaming (p. 57).

Restrict Viewer Access (Use Signed URLs)

If you want requests for objects that match the `PathPattern` for this cache behavior to use public URLs, click **No**.

If you want requests for objects that match the `PathPattern` for this cache behavior to use signed URLs, click **Yes**. Then specify the AWS accounts that you want to use to create signed URLs; these accounts are known as trusted signers.

For more information about trusted signers, see Specifying the AWS Accounts That Can Create Signed URLs (Trusted Signers) (p. 128).

Trusted Signers

Choose which AWS accounts you want to use as trusted signers for this cache behavior:

- **Self:** Use the account with which you're currently signed into the AWS Management Console as a trusted signer. If you're currently signed in as an IAM user, the associated AWS account is added as a trusted signer.
- **Specify Accounts:** Enter account numbers for trusted signers in the **AWS Account Numbers** field.

To create signed URLs, an AWS account must have at least one active CloudFront key pair.

> **Caution**
> If you're updating a distribution that you're already using to distribute content, add trusted signers only when you're ready to start generating signed URLs for your objects. After you add trusted signers to a distribution, users must use signed URLs to access the objects that match the `PathPattern` for this cache behavior.

AWS Account Numbers

If you want to create signed URLs using AWS accounts in addition to or instead of the current account, enter one AWS account number per line in this field. Note the following:

- The accounts that you specify must have at least one active CloudFront key pair. For more information, see Creating CloudFront Key Pairs for Your Trusted Signers (p. 129).
- You can't create CloudFront key pairs for IAM users, so you can't use IAM users as trusted signers.
- For information about how to get the AWS account number for an account, see How Do I Get Security Credentials? in the *Amazon Web Services General Reference*.
- If you enter the account number for the current account, CloudFront automatically checks the **Self** checkbox and removes the account number from the **AWS Account Numbers** list.

Distribution Details

The following values apply to the entire distribution.

Price Class

Choose the price class that corresponds with the maximum price that you want to pay for CloudFront service. By default, CloudFront serves your objects from edge locations in all CloudFront regions.

For more information about price classes and about how your choice of price class affects CloudFront performance for your distribution, see Choosing the Price Class for a CloudFront Distribution (p. 32). For information about CloudFront pricing, including how price classes map to CloudFront regions, go to Amazon CloudFront Pricing.

Alternate Domain Names (CNAMEs)

Optional. Specify one or more domain names that you want to use for URLs for your objects instead of the domain name that CloudFront assigns when you create your distribution. For example, if you want the URL for the object:

```
/images/image.jpg
```

to look like this:

```
http://www.example.com/images/image.jpg
```

instead of like this:

```
http://d111111abcdef8.cloudfront.net/images/image.jpg
```

add a CNAME for `www.example.com`.

> **Important**
> If you add a CNAME for `www.example.com` to your distribution, you also need to create (or update) a CNAME record with your DNS service to route queries for `www.example.com` to `d111111abcdef8.cloudfront.net`. You must have permission to create a CNAME record with the DNS service provider for the domain. Typically, this means that you own the domain, but you may also be developing an application for the domain owner.

For the current limit on the number of alternate domain names that you can add to a distribution, see Amazon CloudFront Limits in the *Amazon Web Services General Reference*. To request a higher limit, go to https://aws.amazon.com/support/createCase?type=service_limit_increase&serviceLimitIncreaseType=cloudfront-distributions.

For more information about alternate domain names, see Using Alternate Domain Names (CNAMEs) (p. 29). For more information about CloudFront URLs, see Format of URLs for CloudFront Objects (p. 72).

SSL Certificate

If you want viewers to use HTTPS to access your objects, choose the applicable SSL certificate:

- **Default CloudFront Certificate (*.cloudfront.net):** If you want to use the CloudFront domain name in the URLs for your objects, such as `https://d111111abcdef8.cloudfront.net/image1.jpg`, choose this option. Also choose this option if you want viewers to use HTTP to access your objects.
- **The name of a certificate in the IAM certificate store:** If you want to use your own domain name in the URLs for your objects, such as `https://example.com/image1.jpg`, choose the applicable certificate that you previously uploaded to the IAM certificate store. For more information, see Using Alternate Domain Names and HTTPS (p. 171).

  If you choose this setting, we recommend that you use only an alternate domain name in your object URLs (https://example.com/logo.jpg). If you use your CloudFront distribution domain name (https://d111111abcdef8.cloudfront.net/logo.jpg) and the viewer supports SNI, then CloudFront behaves normally. However, a viewer that does not support SNI exhibits one of the following behaviors, depending on the value of **Clients Supported**:
  - **All Clients**: If the viewer doesn't support SNI, it displays a warning because the CloudFront domain name doesn't match the domain name in your SSL certificate.
  - **Only Clients that Support Server Name Indication (SNI)**: CloudFront drops the connection with the viewer without returning the object.

Clients Supported

If you specified one or more alternate domain names and you specified an SSL certificate in the IAM certificate store, choose how you want CloudFront to serve HTTPS requests, either a method that works for all clients or one that works for most clients:

- **All Clients:** Any client can access your content. However, you must request permission to use this feature, and you incur additional monthly charges.
- **Only Clients that Support Server Name Indication (SNI):** All modern browsers can access your content because they all support SNI. However, some browsers still in use don't support SNI. Users with these browsers must access your content using some other method, for example, by getting your objects directly from the origin.

For more information, see Using Alternate Domain Names and HTTPS (p. 171).

Default Root Object

Optional. The object that you want CloudFront to request from your origin (for example, `index.html`) when a viewer requests the root URL of your distribution (`http://www.example.com/`) instead of an object in your distribution (`http://www.example.com/product-description.html`). Specifying a default root object avoids exposing the contents of your distribution.

The maximum length of the name is 255 characters. The name can contain any of the following characters:

- A-Z, a-z
- 0-9
- `_ - . * $ / ~ " '`
- &, passed and returned as `&amp;`

When you specify the default root object, enter only the object name, for example, `index.html`. Do not add a `/` before the object name.

For more information, see Specifying a Default Root Object (Web Distributions Only) (p. 96).

Logging

Whether you want CloudFront to log information about each request for an object and store the log files in an Amazon S3 bucket. You can enable or disable logging at any time. There is no extra charge if you enable logging, but you accrue the usual Amazon S3 charges for storing and accessing the files in an Amazon S3 bucket. You can delete the logs at any time. For more information about CloudFront access logs, see Access Logs (p. 182).

Bucket for Logs

If you chose **On** for **Logging**, the Amazon S3 bucket that you want CloudFront to store access logs in, for example, `myawslogbucket.s3.amazonaws.com`. If you enable logging, CloudFront records information about each end-user request for an object and stores the files in the specified Amazon S3 bucket. You can enable or disable logging at any time. For more information about CloudFront access logs, see Access Logs (p. 182).

Log Prefix

Optional. If you chose **On** for **Logging**, specify the string, if any, that you want CloudFront to prefix to the access log filenames for this distribution, for example, `exampleprefix/`. The trailing slash (`/`) is optional but recommended to simplify browsing your log files. For more information about CloudFront access logs, see Access Logs (p. 182).

Cookie Logging

If you want CloudFront to include cookies in access logs, choose **On**. If you choose to include cookies in logs, CloudFront logs all cookies regardless of how you configure the cache behaviors for this distribution: forward all cookies, forward no cookies, or forward a specified list of cookies to the origin.

Amazon S3 doesn't process cookies, so unless your distribution also includes an Amazon EC2 or other custom origin, we recommend that you choose **Off** for the value of **Cookie Logging**.

For more information about cookies, go to Configuring CloudFront to Cache Objects Based on Cookies (p. 76).

Comment

Optional. When you create a distribution, you can include a comment of up to 128 characters. You can update the comment at any time.

Distribution State

Indicates whether you want the distribution to be enabled or disabled once it's deployed:

- *Enabled* means that as soon as the distribution is fully deployed you can deploy links that use the distribution's domain name and end users can retrieve content. Whenever a distribution is enabled, CloudFront accepts and handles any end-user requests for content that use the domain name associated with that distribution.

  When you create, modify, or delete a CloudFront distribution, it takes time for your changes to propagate to the CloudFront database. An immediate request for information about a distribution might not show the change. Propagation usually completes within minutes, but a high system load or network partition might increase this time.
- *Disabled* means that even though the distribution might be deployed and ready to use, end users can't use it. Whenever a distribution is disabled, CloudFront doesn't accept any end-user requests that use the domain name associated with that distribution. Until you switch the distribution from disabled to enabled (by updating the distribution's configuration), no one can use it.

You can toggle a distribution between disabled and enabled as often as you want. Follow the process for updating a distribution's configuration. For more information, see Listing, Viewing, and Updating CloudFront Distributions (p. 27).

Custom Error Pages and Error Caching

You can have CloudFront return an object to the viewer (for example, an HTML file) when your Amazon S3 or custom origin returns an HTTP 4xx or 5xx status code to CloudFront. You can also specify how long an error response from your origin or a custom error page is cached in CloudFront edge caches. For more information, see Customizing Error Responses (p. 92).

> **Note**
> The following values aren't included in the Create Distribution wizard, so you can configure custom error pages only when you update a distribution.

Error Code

The HTTP status code for which you want CloudFront to return a custom error page. You can configure CloudFront to return custom error pages for none, some, or all of the HTTP status codes that CloudFront caches.

Response Page Path

The path to the custom error page (for example, `/4xx-errors/403-forbidden.html`) that you want CloudFront to return to a viewer when your origin returns the HTTP status code that you specified for **Error Code** (for example, 403). If you want to store your objects and your custom error pages in different locations, your distribution must include a cache behavior for which the following is true:

- The value of **Path Pattern** matches the path to your custom error messages. For example, suppose you saved custom error pages for 4xx errors in an Amazon S3 bucket in a directory named `/4xx-errors`. Your distribution must include a cache behavior for which the path pattern routes requests for your custom error pages to that location, for example, **/4xx-errors/***.

- The value of **Origin** specifies the value of **Origin ID** for the origin that contains your custom error pages.

Response Code

The HTTP status code that you want CloudFront to return to the viewer along with the custom error page.

Error Caching Minimum TTL

The minimum amount of time that you want CloudFront to cache error responses from your origin server.

Restrictions

If you need to prevent users in selected countries from accessing your content, you can configure your CloudFront distribution either to allow users in a whitelist of specified countries to access your content or to not allow users in a blacklist of specified countries to access your content. For more information, see Restricting the Geographic Distribution of Your Content (p. 56).

> **Note**
> The following values aren't included in the Create Distribution wizard, so you can configure geo restrictions only when you update a distribution.

Enable Geo Restriction

Whether you want to prevent users in selected countries from accessing your content. There is no additional charge for configuring geo restriction.

Restriction Type

How you want to specify the countries from which your users can access your content:

- **Whitelist:** The **Countries** list includes all of the countries from which you *do* want your users to access your content.
- **Blacklist:** The **Countries** list includes all of the countries from which you *do not* want your users to access your content.

Countries

The countries that you want to add to your whitelist or blacklist. To add a country, select it in the list on the left and click **Add**. Note the following:

- To add multiple consecutive countries, select the first country, press and hold the Shift key, select the last country, and click **Add**.
- To add multiple non-consecutive countries, select the first country, press and hold the Ctrl key, select the remaining countries, and click **Add**.
- To find a country in the left list, enter the first few characters of the country's full name.
- The two-letter code before the name of each country is the value that you enter if you want to create or update a distribution by using the CloudFront API. We use the International Organization for Standardization country codes. For an easy-to-use list, sortable by code and by country name, see the Wikipedia entry ISO 3166-1 alpha-2.

Values that CloudFront Displays in the Console When You Create or Update a Web Distribution

When you create a new web distribution or update an existing distribution, CloudFront displays the following information in the CloudFront console.

Note
Active trusted signers, the AWS accounts that have an active CloudFront key pair and can be used to create valid signed URLs, are currently not visible in the CloudFront console.

Distribution ID (General Tab)

When you perform an action on a distribution using the CloudFront API, you use the distribution ID to specify which distribution you want to perform the action on, for example, `EDFDVBD6EXAMPLE`. You can't change the distribution ID.

Distribution Status (General Tab)

The possible status values for a distribution are listed in the following table.

Value	Description
InProgress	The distribution is still being created or updated.
Deployed	The distribution has been created or updated and the changes have been fully propagated through the CloudFront system.

Note
In addition to ensuring that the status for a distribution is **Deployed**, you must enable the distribution before end users can use CloudFront to access your content. For more information, see Distribution State (p. 53).

Last Modified (General Tab)

The date and time that the distribution was last modified, using ISO 8601 format, for example, 2012-05-19T19:37:58Z. For more information, go to http://www.w3.org/TR/NOTE-datetime.

Domain Name (General Tab)

You use the distribution's domain name in the links to your objects. For example, if your distribution's domain name is `d111111abcdef8.cloudfront.net`, the link to `/images/image.jpg` would be `http://d111111abcdef8.cloudfront.net/images/image.jpg`. You can't change the CloudFront domain name for your distribution. For more information about CloudFront URLs for links to your objects, see Format of URLs for CloudFront Objects (p. 72).

If you specified one or more alternate domain names (CNAMEs), you can use your own domain names for links to your objects instead of using the CloudFront domain name. For more information about CNAMEs, see Alternate Domain Names (CNAMEs) (p. 50).

Note
CloudFront domain names are unique. Your distribution's domain name was never used for a previous distribution and will never be reused for another distribution in the future.

Requirements and Recommendations for Using Amazon EC2 and Other Custom Origins

Follow these guidelines for using Amazon EC2 instances and other custom origins with CloudFront.

- Host and serve the same content on all servers.
- Log the `X-Amz-Cf-Id` header entries on all servers; CloudFront requires this information for debugging.
- Restrict access requests to the HTTP and HTTPS ports that your custom origin listens on.
- Synchronize the clocks of all servers in your implementation.
- Use redundant servers to handle failures.
- For information about request and response behavior and about supported HTTP status codes, see Request and Response Behavior (p. 101).

If you use Amazon Elastic Compute Cloud for your custom origins, we recommend that you do the following:

1. Use an Amazon Machine Image that automatically installs the software for a web server. For more information, see the Amazon EC2 documentation.
2. Use an Elastic Load Balancing load balancer to handle traffic across multiple Amazon EC2 instances and to isolate your application from changes to Amazon EC2 instances. For example, if you use a load balancer, you can add and delete Amazon EC2 instances without changing your application. For more information, see the Elastic Load Balancing documentation.
3. When you create your CloudFront distribution, specify the URL of the load balancer for the domain name of your origin server. For more information, see Working with Web Distributions (p. 36).

Restricting the Geographic Distribution of Your Content

When an end user requests your content, CloudFront typically serves the requested content regardless of where the user is located. If you need to prevent users in selected countries from accessing your content, you can configure a CloudFront web distribution to do one of the following:

- Allow your users to access your content only if they're in a whitelist of specified countries.
- Prevent your users from accessing your content if they're in a blacklist of specified countries.

For example, if a request comes from a country where, for copyright reasons, you are not authorized to distribute your content, you can block the request; this is known as geo restriction or geoblocking.

> **Note**
> CloudFront determines the location of your users using a third-party GeoIP database. The accuracy of the mapping between IP addresses and countries varies by region. Based on recent tests, our overall accuracy is 99.8%.

Here's how geo restriction works:

1. Suppose you only have rights to distribute your content in Lichtenstein. You update your CloudFront web distribution and add a whitelist that contains only Lichtenstein. (Alternatively, you could add a blacklist that contains every country except Lichtenstein.)
2. A user in Monaco requests your content, and DNS routes the request to the CloudFront edge location in Milan, Italy.

3. The edge location in Milan looks up your distribution and determines that the user in Monaco is not allowed to download your content.
4. CloudFront returns an HTTP status code of 403 (forbidden) to the user.

You can optionally configure CloudFront to return a custom error message to the user, and you can specify how long you want CloudFront to cache the error response for the requested object; the default value is five minutes. For more information, see Customizing Error Responses (p. 92).

Geo restriction applies to an entire distribution. If you need to apply one restriction to part of your content and a different restriction (or no restriction) to another part of your content, you must create separate CloudFront web distributions.

If you enable CloudFront access logging, you can identify the requests that CloudFront rejected with an HTTP status code of 403. However, using only the access logs, you can't distinguish a request that CloudFront rejected based on the location of the user from a request that CloudFront rejected because the user didn't have permission to access the object for another reason. If you have a geolocation service such as Digital Element or MaxMind, you can identify the location of requests based on the IP address in the `c-ip` (client IP) column in the access logs. For more information about CloudFront access logs, see Access Logs (p. 182).

> **Note**
> If you need to restrict distribution of your content in geographic regions that don't follow country boundaries, if you want to restrict distribution of individual files, or if you want to build geo restriction into your application, you can combine CloudFront with a third-party service. For more information, see the tutorial Restricting Access to Files in a CloudFront Distribution Based on Geographic Location (Geoblocking) (p. 244).

The following procedure explains how to use the CloudFront console to add geo restriction to an existing web distribution. For information about how to use the console to create a web distribution, see Working with Web Distributions (p. 36).

To use the CloudFront console to add geo restriction to your CloudFront web distribution

1. Sign in to the AWS Management Console and open the Amazon CloudFront console at https://console.aws.amazon.com/cloudfront/.
2. In the top pane of the CloudFront console, select the distribution that you want to update.

   > **Note**
   > The top pane lists all of the distributions that are associated with the AWS account that you used when you signed in to the CloudFront console.

3. In the **Distribution Settings** pane, click the **Restrictions** tab.
4. Click **Edit**.
5. Enter the applicable values. For more information, see Restrictions (p. 54).
6. Click **Yes, Edit**.

Configuring On-Demand Smooth Streaming

You can use CloudFront for on-demand streaming of media files that you've transcoded into the Microsoft Smooth Streaming format. To distribute Smooth Streaming content on demand, you have two options:

- As the origin for your distribution, specify a web server that can stream files that have been transcoded into Microsoft Smooth Streaming format.
- Enable Smooth Streaming in a CloudFront distribution. Smooth Streaming is a property of cache behaviors, which means that you can use one distribution to distribute Smooth Streaming media files as well as other content.

If you enable Smooth Streaming, note the following:

- You can still distribute other content using the same cache behavior if the content matches the value of **Path Pattern** for that cache behavior.
- CloudFront can use either an Amazon S3 bucket or a custom origin for Smooth Streaming media files. However, CloudFront cannot use a Microsoft IIS Server as an origin if the server is configured for Smooth Streaming.
- You cannot invalidate media files in the Smooth Streaming format. If you want to update files before they expire, you must rename them. For more information, see Adding, Removing, or Replacing Objects in a Distribution (p. 81).

For information about Smooth Streaming clients, see Smooth Streaming Primer on the Microsoft website.

To use CloudFront to stream media files that have been encoded in the Microsoft Smooth Streaming format without using a web server that can stream files in Smooth Streaming format, perform the following tasks:

1. Transcode your media files into Smooth Streaming fragmented-MP4 format. For a list of applications that can transcode into Smooth Streaming format, see Smooth Streaming Primer on the Microsoft website.
2. Do one of the following:

   - **If you're using the CloudFront console:** Enable Smooth Streaming by updating the default cache behavior and/or one or more custom cache behaviors in an existing CloudFront web distribution. (You can't configure Smooth Streaming when you create a distribution, so you need to update an existing distribution.) For information about how to update a distribution by using the console, see Listing, Viewing, and Updating CloudFront Distributions (p. 27).
   - **If you're using the CloudFront API:** Add the `SmoothStreaming` element to the `DistributionConfig` complex type for the default cache behavior and/or one or more custom cache behaviors.

3. Upload the files in your Smooth Streaming presentations to the applicable origin.
4. Create either a `clientaccesspolicy.xml` or a `crossdomainpolicy.xml` file, and add it to a location that is accessible at the root of your distribution, for example, `http://d111111abcdef8.cloudfront.net/clientaccesspolicy.xml`. For more information, see Making a Service Available Across Domain Boundaries on the Microsoft Developer Network website.
5. For links in your application, specify the client manifest in the following format:

   ```
   http://d111111abcdef8.cloudfront.net/video/presentation.ism/Manifest
   ```

Configuring On-Demand Progressive Downloads

To use CloudFront to distribute media files using progressive download, perform the following tasks:

1. Transcode your media files, if applicable.
2. Create a CloudFront web distribution. No special settings are required.
3. Upload your files to the origin that you specified when you created your distribution.
4. For links in your application (for example, a media player), specify the name of the media file in the same format that you use for other objects that you're distributing using CloudFront. For more information, see Format of URLs for CloudFront Objects (p. 72).

Configuring On-Demand Apple HTTP Live Streaming (HLS)

To use CloudFront to stream media files in Apple HTTP Live Streaming (HLS) format, perform the following tasks:

1. Transcode your media files into HLS format. You can use any transcoding application that supports HLS format, for example, Amazon Elastic Transcoder.
2. Create a CloudFront web distribution. No special settings are required.
3. Upload your files to the origin that you specified when you created your distribution.
4. For links in your application (for example, a media player), specify the master playlist in the same format that you use for other objects that you're distributing using CloudFront. For more information, see Format of URLs for CloudFront Objects (p. 72).

Working with RTMP Distributions

Topics

- How RTMP Distributions Work (p. 60)
- Task List for Streaming Media Files Using RTMP (p. 62)
- Creating an RTMP Distribution Using the CloudFront Console (p. 63)
- Values that You Specify When You Create or Update an RTMP Distribution (p. 63)
- Values that CloudFront Displays in the Console When You Create or Update an RTMP Distribution (p. 68)
- Configuring the Media Player (p. 69)
- Using an Amazon S3 Bucket as the Origin for an RTMP Distribution (p. 69)
- Creating Multiple RTMP Distributions for an Origin Server (p. 70)
- Restricting Access Using Crossdomain.xml (p. 70)
- Error Codes for RTMP Distributions (p. 71)
- Troubleshooting RTMP Distributions (p. 71)

This section describes how you configure and manage RTMP distributions. For information about how to create an RTMP distribution, see Task List for Streaming Media Files Using RTMP (p. 62).

How RTMP Distributions Work

To stream media files using CloudFront, you provide two types of files to your end users:

- Your media files
- A media player, for example, JW Player, Flowplayer, or Adobe Flash

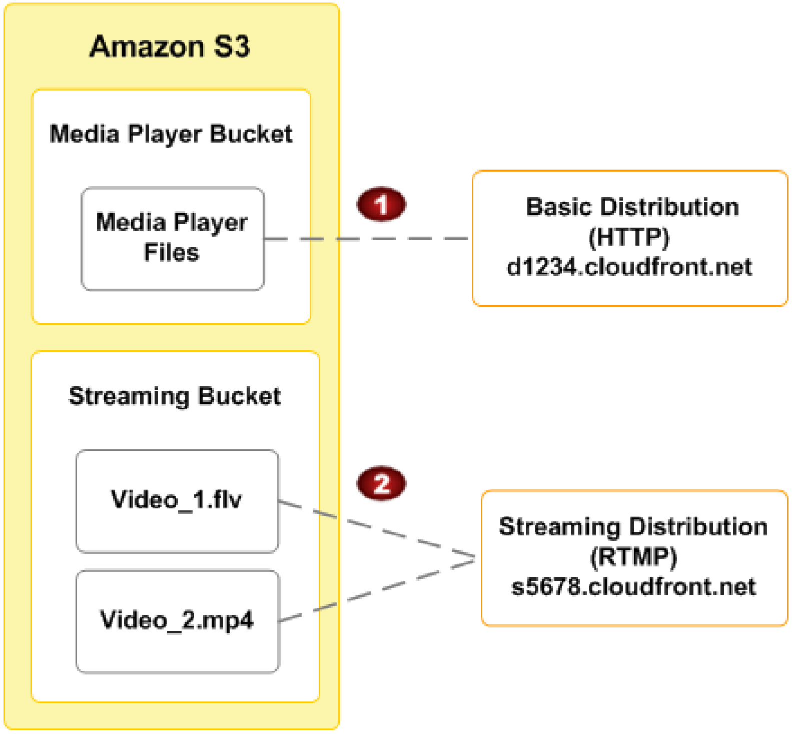
End users view your media files using the media player that you provide for them; they do not use the media player (if any) that is already installed on their computer or other device.

When an end user streams your media file, the media player begins to play the content of the file while the file is still being downloaded from CloudFront. The media file is not stored locally on the end user's system.

To use CloudFront to serve both the media player and the media files, you need two types of distributions: a web distribution for the media player, and an RTMP distribution for the media files. Web distributions serve files over HTTP, while RTMP distributions stream media files over RTMP (or a variant of RTMP).

The following example assumes that your media files and your media player are stored in different buckets in Amazon S3, but that isn't required—you can store media files and your media player in the same Amazon S3 bucket. You can also make the media player available to end users in other ways, for example, using CloudFront and a custom origin. However, the media files must use an Amazon S3 bucket as the origin.

In the following diagram, your site serves a cached copy of the media player to each end user through the `d1234.cloudfront.net` domain. The media player then accesses cached copies of your media files through the `s5678.cloudfront.net` domain.


Amazon S3
Media Player Bucket
Media Player Files
Streaming Bucket
Video_1.flv
Video_2.mp4
1
2
Basic Distribution (HTTP) d1234.cloudfront.net
Streaming Distribution (RTMP) s5678.cloudfront.net

❶	Your media player bucket holds the media player and is the origin server for a regular HTTP distribution. In this example, the domain name for the distribution is `d1234.cloudfront.net`. (The d in `d1234.cloudfront.net` indicates that this is a web distribution.)
❷	Your streaming media bucket holds your media files and is the origin server for an RTMP distribution. In this example, the domain name for the distribution is `s5678.cloudfront.net`. (The s in `s5678.cloudfront.net` indicates that this is an RTMP distribution.)

When you configure CloudFront to distribute media files, CloudFront uses Adobe Flash Media Server 3.5 as the streaming server and streams your media files using Adobe's Real-Time Messaging Protocol (RTMP). CloudFront accepts RTMP requests over port 1935 and port 80.

CloudFront supports the following variants of the RTMP protocol:

- **RTMP—**Adobe's Real-Time Message Protocol
- **RTMPT—**Adobe streaming tunneled over HTTP

- **RTMPE—**Adobe encrypted
- **RTMPTE—**Adobe encrypted tunneled over HTTP

For a basic summary of RTMP and the file formats that Adobe Flash Media Server supports, go to Overview of Streaming with Flash Media Server 3 on the Adobe website. The overview includes information about supported codecs and containers.

Resources available on the Internet can help you determine the bit rate to use for your Flash files, for example, the Flash video (FLV) bitrate calculator on the Adobe website.

CloudFront supports all of the features in Adobe Flash Media Server 3.5 related to *dynamic streaming*, which lets you switch between streams of different qualities during playback. For more information, go to Dynamic streaming in Flash Media Server 3.5: Part 1 on the Adobe website.

To serve streamed content, you need to provide your end users with a media player. You can write your own player using Adobe Flash. For more information, go to http://www.adobe.com/products/flashplayer/. Alternatively, you can use an existing player. For more information, see the following tutorials:

- On-Demand Video Streaming Using CloudFront and Adobe Flash Player (p. 269)
- On-Demand Video Streaming Using CloudFront and Flowplayer for Adobe Flash (p. 274)
- On-Demand Video Streaming Using CloudFront and JW Player (p. 279)

Task List for Streaming Media Files Using RTMP

This section summarizes the general process for configuring on-demand streaming using the Adobe RTMP protocol for any media player. If you're using Adobe Flash Player, Flowplayer, or JW Player for your media player, see the applicable tutorial instead:

- On-Demand Video Streaming Using CloudFront and Adobe Flash Player (p. 269)
- On-Demand Video Streaming Using CloudFront and Flowplayer for Adobe Flash (p. 274)
- On-Demand Video Streaming Using CloudFront and JW Player (p. 279)

The following task list summarizes the process for creating a web distribution.

To Create an RTMP Distribution

1. Create an Amazon S3 bucket for your media files. If you are using a different Amazon S3 bucket for your media player, create an Amazon S3 bucket for the media player files, too.

   The names of your buckets must be all lowercase and cannot contain spaces.
2. Choose and configure a media player to play your media files. For more information, refer to the documentation for the media player.
3. Upload the files for your media player to the origin from which you want CloudFront to get the files. If you are using an Amazon S3 bucket as the origin for the media player, make the files (not the bucket) publicly readable.
4. Create a web distribution for your media player. (You can also use an existing distribution.) For more information, see Task List for Creating a Web Distribution (p. 36).
5. Upload your media files to the Amazon S3 bucket that you created for the media files, and make the content (not the bucket) publicly readable.

   Important
   Media files in a Flash Video container must include the .flv filename extension, or the media will not stream.

You can put media player files and media files in the same bucket.

6. Create an RTMP distribution for your media files:
   - For more information about creating a web distribution using the CloudFront console, see Creating an RTMP Distribution Using the CloudFront Console (p. 63).
   - For information about creating a web distribution using the CloudFront API, go to POST Streaming Distribution in the *Amazon CloudFront API Reference*.
7. Configure your media player. For more information, see Configuring the Media Player (p. 69).

If you have trouble getting your content to play, see Troubleshooting RTMP Distributions (p. 71).

Creating an RTMP Distribution Using the CloudFront Console

The following procedure explains how to create an RTMP distribution using the CloudFront console. If you want to create an RTMP distribution using the CloudFront API, go to POST Streaming Distribution in the *Amazon CloudFront API Reference*.

For the current limit on the number of RTMP distributions that you can create for each AWS account, see Amazon CloudFront Limits in the *Amazon Web Services General Reference*. To request a higher limit, go to https://aws.amazon.com/support/createCase?type=service_limit_increase&serviceLimitIncreaseType=cloudfront-distributions.

To create an RTMP distribution using the CloudFront console

1. Sign in to the AWS Management Console and open the Amazon CloudFront console at https://console.aws.amazon.com/cloudfront/.
2. Click **Create Distribution**.
3. On the first page of the **Create Distribution Wizard**, click **RTMP**, and click **Continue**.
4. Specify settings for the distribution. For more information, see Values that You Specify When You Create or Update an RTMP Distribution (p. 63).
5. Click **Create Distribution**.
6. After CloudFront creates your distribution, the value of the **Status** column for your distribution will change from **InProgress** to **Deployed**. If you chose to enable the distribution, it will then be ready to process requests. This should take less than 15 minutes.

   The domain name that CloudFront assigns to your distribution appears in the list of distributions. The domain name also appears on the **General** tab for a selected distribution.

Values that You Specify When You Create or Update an RTMP Distribution

To stream media files using CloudFront, you create an RTMP distribution and specify the following values.

Topics

- Origin Domain Name (Amazon S3 Bucket) (p. 64)
- Restrict Bucket Access (Amazon S3 Only) (p. 65)

- Origin Access Identity (Amazon S3 Only) (p. 65)
- Comment for New Identity(Amazon S3 Only) (p. 65)
- Your Identities (Amazon S3 Only) (p. 65)
- Grant Read Permissions on Bucket (Amazon S3 Only) (p. 65)
- Price Class (p. 65)
- Alternate Domain Names (CNAMEs) (p. 66)
- Logging (p. 66)
- Bucket for Logs (p. 66)
- Log Prefix (p. 66)
- Comment (p. 66)
- Distribution State (p. 66)
- Restrict Viewer Access (Use Signed URLs) (p. 67)
- Trusted Signers (p. 67)
- AWS Account Numbers (p. 67)

Origin Domain Name (Amazon S3 Bucket)

The DNS domain name of the Amazon S3 bucket from which you want CloudFront to get objects for this origin, for example, `myawsbucket.s3.amazonaws.com`. In the CloudFront console, click in the **Origin Domain Name** field, and a list enumerates the Amazon S3 buckets that are associated with the current AWS account. To use a bucket from a different AWS account, type the domain name of the bucket in the following format:

```
bucket-name.s3.amazonaws.com
```

If your bucket is in the US Standard region and you want Amazon S3 to route requests to a facility in Northern Virginia, use the following format:

```
bucket-name.s3-external-1.amazonaws.com
```

The files must be publicly readable unless you secure your content in Amazon S3 by using a CloudFront origin access identity. For more information, see Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content (p. 123).

> **Important**
> The bucket name must conform to DNS naming requirements. For more information, go to Bucket Restrictions and Limitations in the *Amazon Simple Storage Service Developer Guide*.

When you change the bucket from which CloudFront gets objects for the current origin, CloudFront immediately begins replicating the change to CloudFront edge locations. Until the distribution configuration is updated in a given edge location, CloudFront will continue to forward requests to the previous Amazon S3 bucket. As soon as the distribution configuration is updated in that edge location, CloudFront begins to forward requests to the new Amazon S3 bucket.

Changing the bucket does not require CloudFront to repopulate edge caches with objects from the new origin. As long as the viewer requests in your application have not changed, CloudFront will continue to serve objects that are already in an edge cache until the TTL on each object expires or until seldom-requested objects are evicted.

For more information, see Using an Amazon S3 Bucket as the Origin for an RTMP Distribution (p. 69).

Restrict Bucket Access (Amazon S3 Only)

Click **Yes** if you want to require end users to access objects in an Amazon S3 bucket by using only CloudFront URLs, not by using Amazon S3 URLs. Then specify the applicable values.

Click **No** if you want end users to be able to access objects using either CloudFront URLs or Amazon S3 URLs.

For more information, see Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content (p. 123).

Origin Access Identity (Amazon S3 Only)

If you chose **Yes** for **Restrict Bucket Access**, choose whether to create a new origin access identity or use an existing one that is associated with your AWS account. If you already have an origin access identity, we recommend that you reuse it to simplify maintenance. For more information about origin access identities, see Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content (p. 123).

Comment for New Identity(Amazon S3 Only)

If you chose **Create a New Identity** for **Origin Access Identity**, enter a comment that identifies the new origin access identity. CloudFront will create the origin access identity when you create this distribution.

Your Identities (Amazon S3 Only)

If you chose **Use an Existing Identity** for **Origin Access Identity**, choose the origin access identity that you want to use. You cannot use an origin access identity that is associated with another AWS account.

Grant Read Permissions on Bucket (Amazon S3 Only)

If you want CloudFront to automatically grant the origin access identity the permission to read objects in your Amazon S3 bucket, click **Yes, Update Bucket Policy**.

> **Important**
> If you click **Yes, Update Bucket Policy**, CloudFront updates the bucket policy to grant the specified origin access identity the permission to read objects in your bucket. However, CloudFront does not remove existing permissions in the bucket policy or permissions on individual objects. If users currently have permission to access the objects in your bucket using Amazon S3 URLs, they will still have that permission after CloudFront updates your bucket policy. To view or change the existing bucket policy and the existing permissions on the objects in your bucket, use a method provided by Amazon S3. For more information, see Granting the Origin Access Identity Permission to Read Objects in Your Amazon S3 Bucket (p. 125).

If you want to update permissions manually, for example, if you want to update ACLs on your objects instead of updating bucket permissions, click **No, I will Update Permissions**.

Price Class

The price class that corresponds with the maximum price that you want to pay for CloudFront service. By default, CloudFront serves your objects from edge locations in all CloudFront regions.

For more information about price classes and about how your choice of price class affects CloudFront performance for your distribution, see Choosing the Price Class for a CloudFront Distribution (p. 32). For information about CloudFront pricing, including how price classes map to CloudFront regions, go to Amazon CloudFront Pricing.

Alternate Domain Names (CNAMEs)

Optional. You can associate one or more CNAME aliases with a distribution so that you can use your domain name (for example, example.com) in the URLs for your objects instead of using the domain name that CloudFront assigned when you created your distribution. For more information, see Using Alternate Domain Names (CNAMEs) (p. 29).

Logging

Whether you want CloudFront to log information about each request for an object and store the log files in an Amazon S3 bucket. You can enable or disable logging at any time. There is no extra charge if you enable logging, but you accrue the usual Amazon S3 charges for storing and accessing the files in an Amazon S3 bucket. You can delete the logs at any time. For more information about CloudFront access logs, see Access Logs (p. 182).

Bucket for Logs

If you chose **On** for **Logging**, the Amazon S3 bucket that you want CloudFront to store access logs in, for example, `myawslogbucket.s3.amazonaws.com`. If you enable logging, CloudFront records information about each end-user request for an object and stores the files in the specified Amazon S3 bucket. You can enable or disable logging at any time. For more information about CloudFront access logs, see Access Logs (p. 182).

Log Prefix

Optional. If you chose **On** for **Logging**, specify the string, if any, that you want CloudFront to prefix to the access log filenames for this distribution, for example, `exampleprefix/`. The trailing slash (`/`) is optional but recommended to simplify browsing your log files. For more information about CloudFront access logs, see Access Logs (p. 182).

Comment

Optional. When you create a distribution, you can include a comment of up to 128 characters. You can update the comment at any time.

Distribution State

When you create a distribution, you must specify whether you want the distribution to be enabled or disabled after it's created:

- *Enabled* means that as soon as the distribution is fully deployed you can deploy links that use the distribution's domain name and end users can retrieve content. Whenever a distribution is enabled, CloudFront accepts and processes any end-user requests for content that use the domain name associated with that distribution.

  When you create, modify, or delete a CloudFront distribution, it takes time for your changes to propagate to the CloudFront database. An immediate request for information about a distribution might not show the change. Propagation usually completes within minutes, but a high system load or network partition might increase this time.
- *Disabled* means that even though the distribution might be deployed and ready to use, end users can't use it. When a distribution is disabled, CloudFront doesn't accept any end-user requests that use the

domain name associated with that distribution. Until you switch the distribution from disabled to enabled (by updating the distribution's configuration), no one can use it.

You can toggle a distribution between disabled and enabled as often as you want. For information about updating a distribution's configuration, see Listing, Viewing, and Updating CloudFront Distributions (p. 27).

Restrict Viewer Access (Use Signed URLs)

If you want requests for objects served by this distribution to use public URLs, click **No**. If you want requests to use signed URLs, click **Yes**. Then specify the AWS accounts that you want to use to create signed URLs; these accounts are known as trusted signers.

For more information about trusted signers, see Specifying the AWS Accounts That Can Create Signed URLs (Trusted Signers) (p. 128).

Trusted Signers

Choose which AWS accounts you want to use as trusted signers for this distribution:

- **Self:** Use the account with which you're currently signed into the AWS Management Console as a trusted signer. If you're currently signed in as an IAM user, the associated AWS account is added as a trusted signer.
- **Specify Accounts:** Enter account numbers for trusted signers in the **AWS Account Numbers** field.

To create signed URLs, an AWS account must have at least one active CloudFront key pair.

> **Caution**
> If you're updating a distribution that you're already using to distribute content, add trusted signers only when you're ready to start generating signed URLs for your objects. After you add trusted signers to a distribution, users must use signed URLs to access the objects served by this distribution.

AWS Account Numbers

If you want to create signed URLs using AWS accounts in addition to or instead of the current account, enter one AWS account number per line in this field. Note the following:

- The accounts that you specify must have at least one active CloudFront key pair. For more information, see Creating CloudFront Key Pairs for Your Trusted Signers (p. 129).
- You can't create CloudFront key pairs for IAM users, so you can't use IAM users as trusted signers.
- For information about how to get the AWS account number for an account, see How Do I Get Security Credentials? in the *Amazon Web Services General Reference*.
- If you enter the account number for the current account, CloudFront automatically checks the **Self** checkbox and removes the account number from the **AWS Account Numbers** list.

Values that CloudFront Displays in the Console When You Create or Update an RTMP Distribution

When you create a new RTMP distribution or update an existing distribution, CloudFront displays the following information in the CloudFront console.

> **Note**
> Active trusted signers, the AWS accounts that have an active CloudFront key pair and can be used to create valid signed URLs, are currently not visible in the CloudFront console.

Distribution ID

When you perform an action on a distribution using the CloudFront API, you use the distribution ID to specify which distribution you want to perform the action on, for example, `EDFDVBD6EXAMPLE`. You can't change the distribution ID.

Status

The possible status values for a distribution are listed in the following table.

Value	Description
InProgress	The distribution is still being created or updated.
Deployed	The distribution has been created or updated and the changes have been fully propagated through the CloudFront system.

In addition to ensuring that the status for a distribution is **Deployed**, you must enable the distribution before end users can use CloudFront to access your content. For more information, see Distribution State (p. 66).

Last Modified

The date and time that the distribution was last modified, using ISO 8601 format, for example, 2012-05-19T19:37:58Z. For more information, go to http://www.w3.org/TR/NOTE-datetime.

Domain Name

You use the distribution's domain name in the links to your objects, unless you're using alternate domain names (CNAMEs). For example, if your distribution's domain name is `d111111abcdef8.cloudfront.net`, the link to the example `/images/image.jpg` file would be `http://d111111abcdef8.cloudfront.net/images/image.jpg`. You can't change the CloudFront domain name for your distribution. For more information about CloudFront URLs for links to your objects, see Format of URLs for CloudFront Objects (p. 72).

If you specified one or more alternate domain names (CNAMEs), you can use your own domain names for links to your objects instead of using the CloudFront domain name. For more information about CNAMEs, see Alternate Domain Names (CNAMEs) (p. 50).

Note
CloudFront domain names are unique. Your distribution's domain name was never used for a previous distribution and will never be reused for another distribution in the future.

Configuring the Media Player

To play a media file, you must configure the media player with the correct path to the file. How you configure the media depends on which media player you're using and how you're using it.

When you configure the media player, the path you specify to the media file must contain the characters `cfx/st` immediately after the domain name, for example:

`rtmp://s5c39gqb8ow64r.cloudfront.net/cfx/st/mediafile.flv`.

Note
CloudFront follows Adobe's FMS naming requirements. Different players have their own rules about how to specify streams. The example above is for JW Player. Check your player's documentation. For example, Adobe's Flash Media Server does not allow the `.flv` extension to be present on the play path. Many players remove the `.flv` extension for you.

Your media player might ask for the path separate from the file name. For example, with the JW Player wizard, you specify a `streamer` and `file` variable:

- **streamer—**`rtmp://s5c39gqb8ow64r.cloudfront.net/cfx/st` (with no trailing slash)
- **file—** `mediafile.flv`

If you've stored the media files in a directory in your bucket (for example, `videos/mediafile.flv`), then the variables for JW Player would be:

- **streamer—**`rtmp://s5c39gqb8ow64r.cloudfront.net/cfx/st` (with no trailing slash)
- **file—** `videos/mediafile.flv`

To use the JW Player wizard, go to the Setup Wizard page on the JW Player website.

MPEG Files

To serve MP3 audio files or H.264/MPEG-4 video files, you might need to prefix the file name with `mp3:` or `mp4:`. Some media players can be configured to add the prefix automatically. The media player might also require you to specify the file name without the file extension (for example, `magicvideo` instead of `magicvideo.mp4`).

Using an Amazon S3 Bucket as the Origin for an RTMP Distribution

When you create a distribution, you specify where CloudFront gets the files that it distributes to edge locations. For an RTMP distribution, you must use an Amazon S3 bucket; custom origins are not supported. To get your objects into your bucket, you can use any method supported by Amazon S3, for example, the Amazon S3 API or a third-party tool. You can create a hierarchy in your bucket just as you would with any other Amazon S3 bucket. You incur regular Amazon S3 charges for storing the objects in the bucket. For more information about the charges to use CloudFront, see CloudFront Billing and Usage Reports (p. 7).

Using an existing Amazon S3 bucket as your CloudFront origin server doesn't change the bucket in any way; you can still use it as you normally would to store and access Amazon S3 objects (at the normal Amazon S3 prices).

You can use the same Amazon S3 bucket for both RTMP and web distributions.

Note
After you create an RTMP distribution, you can't change its origin server. If you need to change the Amazon S3 bucket for an RTMP distribution, you must create a new distribution that uses the new bucket and update either your links or your DNS records to use the domain name for the new distribution. You can then delete the original distribution. For more information, see Deleting a Distribution (p. 28).

When you specify the name of the Amazon S3 bucket that you want CloudFront to get objects from, you generally use the following format:

```
bucket-name.s3.amazonaws.com
```

If your bucket is in the US Standard region and you want Amazon S3 to route requests to a facility in Northern Virginia, use the following format:

```
bucket-name.s3-external-1.amazonaws.com
```

Do not specify the name of the bucket using the following values:

- The Amazon S3 path style, `s3.amazonaws.com/bucket-name`
- The Amazon S3 CNAME, if any

Important
For your bucket to work with CloudFront, the name must conform to DNS naming requirements. For more information, go to Bucket Restrictions and Limitations in the *Amazon Simple Storage Service Developer Guide*.

Creating Multiple RTMP Distributions for an Origin Server

You typically create one RTMP distribution per Amazon S3 bucket, but you can choose to create multiple RTMP distributions for the same bucket. For example, if you had two distributions for an Amazon S3 bucket, you could reference a single media file using either distribution. In this case, if you had a media file called `media.flv` in your origin server, CloudFront would work with each distribution as though it referenced an individual `media.flv` object: one `media.flv` accessible through one distribution, and another `media.flv` accessible through the other distribution.

Restricting Access Using Crossdomain.xml

The Adobe Flash Media Server `crossdomain.xml` file specifies which domains can access media files in a particular domain. CloudFront supplies a default file that allows all domains to access the media files in your RTMP distribution, and you cannot change this behavior. If you include a more restrictive `crossdomain.xml` file in your Amazon S3 bucket, CloudFront ignores it.

Error Codes for RTMP Distributions

The following table lists the error codes that CloudFront can send to your media player. The errors are part of the string returned with `Event.info.application.message` or `Event.info.description`.

Error	Description
`DistributionNotFound`	The distribution was not found.
`DistributionTypeMismatch`	The distribution is not an RTMP distribution.
`InvalidInstance`	The instance is invalid.
`InvalidURI`	The URI is invalid.

Troubleshooting RTMP Distributions

If you're having trouble getting your media files to play, check the following items.

Item to Check	Description
Separate distributions for the media player files and media files	The media player must be served by a regular HTTP distribution (for example, domain name d111111abcdef8.cloudfront.net), and media files must be served by an RTMP distribution (for example, domain name s5c39gqb8ow64r.cloudfront.net). Make sure you're not using the same distribution for both.
`/cfx/st` in the file path	Confirm that the path for the file includes `/cfx/st`. You don't need to include `/cfx/st` in the path to the object in the Amazon S3 bucket. For more information, see Configuring the Media Player (p. 69).
File names in the file path	Some media players require that you include the file name extension (for example, `mp4:`) before the file name in the file path. Some media players also require that you exclude the file name extension (for example, `.mp4`) from the file path. For more information, see MPEG Files (p. 69). **Note** The names of the media files in your Amazon S3 bucket must always include the applicable file name extension.
Port 1935 on your firewall	Adobe Flash Media Server uses port 1935 for RTMP. Make sure your firewall has this port open. If it doesn't, the typical message returned is "Unable to play video." You can also switch to RTMPT to tunnel over HTTP using port 80.
Adobe Flash Player messaging	By default, the Adobe Flash Player won't display a message if the video file it's trying to play is missing. Instead, it waits for the file to show up. You might want to change this behavior to give your end users a better experience. To instead have the player send a message if the video is missing, use `play("vid",0,-1)` instead of `play("vid")`.

Working with Objects

Topics

- Format of URLs for CloudFront Objects (p. 72)
- How CloudFront Processes HTTP and HTTPS Requests (p. 74)
- Configuring CloudFront to Cache Based on Query String Parameters (p. 75)
- Configuring CloudFront to Cache Objects Based on Cookies (p. 76)
- Configuring CloudFront to Cache Objects Based on Request Headers (p. 78)
- Adding, Removing, or Replacing Objects in a Distribution (p. 81)
- Customizing Error Responses (p. 92)
- How CloudFront Processes Partial Requests for an Object (Range GETs) (p. 96)
- Specifying a Default Root Object (Web Distributions Only) (p. 96)
- Serving Compressed Files (p. 98)

This section describes how you work with objects in CloudFront.

Format of URLs for CloudFront Objects

Topics

- Format of Public URLs for Objects in Amazon S3 (p. 73)
- Format of Public URLs for Objects in a Custom Origin (p. 73)
- How Public URLs Affect the Invalidation of Directories (p. 74)
- Format of Signed URLs (p. 74)

When you create a distribution, you receive the CloudFront domain name associated with that distribution. You use that domain name when creating the links to your objects. If you have another domain name that you'd rather use (for example, `www.example.com`), you can add a CNAME alias. For more information, see Using Alternate Domain Names (CNAMEs) (p. 29).

When you create URLs to give end users access to objects in your CloudFront distribution, the URLs are either public URLs or signed URLs:

Public URLs allow users to access the following objects:

- Objects on which there are no restrictions.
- Objects in an Amazon S3 bucket that end users must access through CloudFront but that don't require a signed URL. These objects can't be accessed using an Amazon S3 URL.

Signed URLs are required to access the objects that are specified by a cache behavior that you have configured to require signed URLs. Note that if a request for an object (for example, `image.jpg`) matches the path patterns for two or more cache behaviors, CloudFront will process the request based on the cache behavior that is listed first in the distribution. If the first cache behavior doesn't require signed URLs and the second cache behavior does require signed URLs, end users will be able to access `image.jpg` without a signed URL.

For more information about cache behaviors, including path patterns, see Cache Behavior Settings (p. 44). For more information about signed URLs, see Serving Private Content through CloudFront (p. 118).

Format of Public URLs for Objects in Amazon S3

A public URL for an object in an Amazon S3 bucket uses this format:

```
http://<CloudFront domain name>/<object name in Amazon S3 bucket>
```

Important
If the distribution serves streaming content, additional characters are required in the path to the file. For more information, see Configuring the Media Player (p. 69).

For example, suppose you have an Amazon S3 bucket called `mybucket`. The bucket contains a publicly readable object named `/images/image.jpg`.

You create a CloudFront distribution and specify `mybucket.s3.amazonaws.com` as the origin server for this distribution. CloudFront returns `d111111abcdef8.cloudfront.net` as the domain name for the distribution and `EDFDVBD6EXAMPLE` as the distribution ID.

The URL you give to end users to access the object in this example is:

`http://d111111abcdef8.cloudfront.net/images/image.jpg`.

For web distributions, if you're storing your content in more than one Amazon S3 bucket, the format of URLs is the same—URLs don't include any information about your Amazon S3 buckets. To route requests to the applicable bucket, you create an origin for each bucket and create one or more cache behaviors that route requests to each origin. The path pattern in a cache behavior specifies which requests are routed to the origin (the Amazon S3 bucket) that is associated with that cache behavior. For more information about the settings for origins and for cache behaviors in a CloudFront distribution, see Values that You Specify When You Create or Update a Web Distribution (p. 40).

For more information about names and paths for Amazon S3 buckets, see Virtual Hosting of Buckets in the *Amazon Simple Storage Service Developer Guide*.

Anytime an end user accesses that object, CloudFront serves the object from the appropriate edge location. If the object isn't in that edge location, CloudFront goes to the origin server associated with the `EDFDVBD6EXAMPLE` distribution (`mybucket.s3.amazonaws.com`) and gets a copy of the object for the edge location to serve to the end user.

Format of Public URLs for Objects in a Custom Origin

The format of public URLs for objects in a custom origin are much like public URLs for objects in Amazon S3:

```
http://<CloudFront domain name>/<object name in custom origin>
```

If your object is in a folder on your origin server, then the CloudFront URL must include the name of the folder. For example, if `image.jpg` is located in the `images` folder, then the URL is:

```
http://d111111abcdef8.cloudfront.net/images/image.jpg
```

CloudFront gets objects from the domain that you specified when you created the distribution, using the object path and name in the public URL. For example, if the domain for your custom origin is `example.com` and the object path and name is `/images/image.jpg`, CloudFront will get the object from the following location:

```
http://example.com/images/image.jpg
```

If you're storing your content on more than one custom origin, the format of URLs is the same—URLs don't include any information about the custom origin. To route requests to the applicable custom origin, you add an origin to your distribution for each custom origin and create one or more cache behaviors that route requests to each origin. The path pattern in a cache behavior specifies which requests are routed to the origin that is associated with that cache behavior. For more information about the settings for origins and for cache behaviors in a CloudFront distribution, see Values that You Specify When You Create or Update a Web Distribution (p. 40).

How Public URLs Affect the Invalidation of Directories

If you use CloudFront URLs that give end users access to directories, we recommend that you always use the same URL format, either with a trailing slash (/) or without, for example:

```
http://d111111abcdef8.cloudfront.net/images/
```

```
http://d111111abcdef8.cloudfront.net/images
```

Browsers and other web applications will resolve both formats to the same directory. However, CloudFront stores public URLs exactly as you specify them, and if you want to invalidate a directory, you'll need to specify the exact same directory, including or excluding the slash. If you don't have a standard for how you specify directories, you'll need to invalidate the directory with and without the slash to ensure that CloudFront removes the directory from the edge location. If you've reached the limit for free invalidations for the month, you'll pay for both invalidations even though only one of the directories exists.

Format of Signed URLs

Signed URLs allow end users to access objects in a distribution that is configured to serve private content. The URLs include extra information that restricts access to the cached objects. For information about the format of signed URLs, see Serving Private Content through CloudFront (p. 118).

How CloudFront Processes HTTP and HTTPS Requests

For Amazon S3 origins, CloudFront accepts requests in both HTTP and HTTPS protocols for objects in a CloudFront distribution by default. CloudFront then forwards the requests to your Amazon S3 bucket using the same protocol in which the requests were made.

For custom origins, when you create your distribution, you can specify how CloudFront accesses your origin: HTTP only, or matching the protocol that is used by the viewer. For more information about how CloudFront handles HTTP and HTTPS requests for custom origins, see Protocols (p. 111).

For information about how to restrict your web distribution so that end users can only access objects using HTTPS, see Using an HTTPS Connection to Access Your Objects (p. 168). (This option doesn't apply to RTMP distributions, which use the RTMP protocol.)

> **Note**
> The charge for HTTPS requests is higher than the charge for HTTP requests. For more information about billing rates, go to the CloudFront pricing plan.

Configuring CloudFront to Cache Based on Query String Parameters

For web distributions, you can choose whether you want CloudFront to forward query string parameters to your origin. For RTMP distributions, you cannot configure CloudFront to forward query string parameters to your origin.

For both types of distributions, if you enable logging, CloudFront logs the full URL, including query string parameters. For web distributions, this is true regardless of whether you have configured CloudFront to forward query strings. For more information about CloudFront logging, see Access Logs (p. 182).

For more information, see the applicable topic:

- Query String Parameters and Web Distributions (p. 75)
- Query String Parameters and RTMP Distributions (p. 76)

Query String Parameters and Web Distributions

For web distributions, you can specify whether you want CloudFront to include query strings when it forwards requests to your origin. For example, you can specify whether you want CloudFront to forward the *?parameter1=a* part of the following URL:

http://d111111abcdef8.cloudfront.net/images/image.jpg*?parameter1=a*

If you configure CloudFront to forward query strings to your origin, CloudFront will include the query string portion of the URL when caching the object. For example, the following query strings cause CloudFront to cache three objects. This is true even if your origin always returns the same `image.jpg` regardless of the query string:

- `http://d111111abcdef8.cloudfront.net/images/image.jpg?parameter1=a`
- `http://d111111abcdef8.cloudfront.net/images/image.jpg?parameter1=b`
- `http://d111111abcdef8.cloudfront.net/images/image.jpg?parameter1=c`

If your origin returns different versions of an object (for example, `/images/image.jpg`) based on the query string, select **Yes** for **Forward Query Strings** in the CloudFront console or specify `true` for the value of the `QueryString` element in the `DistributionConfig` complex type when you're using the CloudFront API.

If your origin returns the same version of an object regardless of the query string, select **No** or `false`. This increases the likelihood that CloudFront can serve a request from the cache, which improves performance and reduces the load on your origin.

The order of parameters matters in query strings. If you configure CloudFront to forward query strings to your origin, the following query strings cause CloudFront to cache two objects:

- `http://d111111abcdef8.cloudfront.net/images/image.jpg`*`?parameter1=a&parameter2=b`*
- `http://d111111abcdef8.cloudfront.net/images/image.jpg`*`?parameter2=b&parameter1=a`*

Case also matters in query strings. If you configure CloudFront to forward query strings to your origin, the following query strings cause CloudFront to cache two objects:

- `http://d111111abcdef8.cloudfront.net/images/image.jpg`*`?parameter1=a`*
- `http://d111111abcdef8.cloudfront.net/images/image.jpg`*`?parameter1=A`*

If you're using signed URLs to restrict access to your content (if you added trusted signers to your distribution), CloudFront removes the following query string parameters before forwarding the rest of the URL to your origin:

- `Expires`
- `Key-Pair-Id`
- `Policy`
- `Signature`

This means that if you're using signed URLs and you're configuring CloudFront to forward query string parameters to your origin, your own query string parameters cannot be named `Expires`, `Key-Pair-Id`, `Policy`, or `Signature`.

Query String Parameters and RTMP Distributions

For RTMP distributions, when CloudFront requests an object from the origin server, it removes any query string parameters. For example, if CloudFront receives the following request and `media.flv` is not already in the CloudFront cache:

`http://d111111abcdef8.cloudfront.net/media/media.flv`*`?parameter1=a`*

it sends the following URL to your origin server:

`http://d111111abcdef8.cloudfront.net/media/media.flv`

Configuring CloudFront to Cache Objects Based on Cookies

For HTTP and HTTPS web distributions, you can choose whether you want CloudFront to forward cookies to your origin. For RTMP distributions, you cannot configure CloudFront to process cookies.

> **Important**
> Amazon S3 and some HTTP servers do not process cookies. Do not configure CloudFront cache behaviors to forward cookies to an origin that doesn't process cookies, or you'll adversely affect cacheability and, therefore, performance.

Cookies and Web Distributions

For web distributions, you can choose whether you want CloudFront to forward cookies to your origin and, if so, you can choose whether you want CloudFront to forward all cookies or only the cookies that you specify. If you choose to forward only selected cookies, you specify a whitelist of cookies to forward. Cookie settings affect only which cookies CloudFront forwards to your origin; if your origin returns cookies to CloudFront, CloudFront forwards all of them to the viewer regardless of the cookie settings in your distribution.

For the current limit on the number of cookie names that you can whitelist for each cache behavior, see Amazon CloudFront Limits in the *Amazon Web Services General Reference*. To request a higher limit, go to https://aws.amazon.com/support/createCase?type=service_limit_increase&serviceLimitIncreaseType=cloudfront-distributions.

> **Note**
> If you don't configure CloudFront to forward cookies to your origin, CloudFront removes the `Cookie` header from requests that it forwards to your origin and removes the `Set-Cookie` header from responses that it returns to your clients.

When you configure CloudFront to forward cookies to your origin:

- CloudFront includes cookie name-value pairs, but not cookie attributes (for example, path or expiration time), when it forwards requests to your origin. For example, in the following `Cookie` header, CloudFront forwards the cookie name-value pair `country=ata` but does not forward the cookie attribute `$Path=/`:

  ```
  Cookie: country=ata; $Path=/;
  ```
- If you configure CloudFront to forward only a specified list of cookies, CloudFront removes any cookies not on that list before forwarding the request to your origin.
- `If-Modified-Since` and `If-None-Match` conditional requests are not supported.
- CloudFront sorts the cookies in natural order by cookie name before forwarding the request to your origin.

When your origin returns cookies to CloudFront:

- If you configure CloudFront to forward all cookies to your origin for a given cache behavior, CloudFront caches the response and all cookies and cookie attributes that are returned by your origin. CloudFront also returns the object and all cookies and cookie attributes to the viewer.
- If you configure CloudFront to forward only specified cookies to your origin for a given cache behavior, CloudFront caches the response, the specified cookies, and the associated cookie attributes. CloudFront also returns the object, including the specified cookies and cookie attributes, to the viewer.

  If your origin returns both whitelisted cookies and cookies that aren't on your whitelist, CloudFront caches only the whitelisted cookies. However, it returns all of the cookies to the viewer, whether or not they appear on your whitelist.
- If you do not want CloudFront to cache cookies and cookie attributes, configure your origin server to add the following header in responses to CloudFront:

  ```
  no-cache="Set-Cookie"
  ```

If you configure CloudFront to forward cookies to your origin, CloudFront uses the cookies that the viewer includes in the request to uniquely identify an object in the cache. For example, if three requests for the object `/images/image1.jpg` contain three different name-value pairs, CloudFront will cache the same object three times, once for each name-value pair. This is true even if your origin ignores the cookie values in the request and always returns the same `image1.jpg` object to CloudFront. As a result, CloudFront forwards more requests to your origin server for the same object, which slows performance and increases the load on your origin server. If your origin server does not vary its response based on the value of a given cookie, we recommend that you not configure CloudFront to forward that cookie to your origin.

Cookie names and values are both case sensitive. For example, if two cookies for the same object are identical except for case, CloudFront will cache the object twice.

If you configure CloudFront to log requests and to log cookies, CloudFront logs all cookies and all cookie attributes even if you configure CloudFront not to forward cookies to your origin or you configure CloudFront to forward only a specified list of cookies. For more information about CloudFront logging, see Access Logs (p. 182).

For information about using the CloudFront console to update a distribution so CloudFront forwards cookies to the origin, see Listing, Viewing, and Updating CloudFront Distributions (p. 27). For information about using the CloudFront API to update a distribution, go to PUT Distribution Config in the *Amazon CloudFront API Reference*.

Cookies and RTMP Distributions

For RTMP distributions, when CloudFront requests an object from the origin server, it removes any cookies before forwarding the request to your origin. If your origin returns any cookies along with the object, CloudFront removes them before returning the object to the viewer. For RTMP distributions, CloudFront does not cache cookies in edge caches.

You cannot configure CloudFront to log cookies for RTMP distributions.

Configuring CloudFront to Cache Objects Based on Request Headers

For web distributions, CloudFront lets you choose whether you want CloudFront to forward headers to your origin and to cache separate versions of a specified object based on the header values in viewer requests. This allows you to serve different versions of your content based on the device the user is using, the location of the viewer, the language the viewer is using, and a variety of other criteria. For RTMP distributions, you cannot configure CloudFront to cache based on header values.

Topics

- Headers and Web Distributions (p. 78)
- Headers and RTMP Distributions (p. 81)

Headers and Web Distributions

By default, CloudFront doesn't consider headers when caching your objects in edge locations. If your origin returns two objects and they differ only by the values in the request headers, CloudFront caches only one version of the object.

You can configure CloudFront to forward headers to the origin, which causes CloudFront to cache multiple versions of an object based on the values in one or more request headers. For example, suppose viewer requests for `logo.jpg` contain a custom `Product` header that has a value of either `Acme` or `Apex`. When you configure CloudFront to cache your objects based on the value of the `Product` header, CloudFront forwards requests for `logo.jpg` to the origin and includes the `Product` header and header values. CloudFront caches `logo.jpg` once for requests in which the value of the `Product` header is `Acme` and once for requests in which the value is `Apex`.

You can configure each cache behavior in a web distribution to do one of the following:

- Forward all headers to your origin

  Important
  If you configure CloudFront to forward all headers to your origin, CloudFront doesn't cache the objects associated with this cache behavior. Instead, it sends every request to the origin.
- Forward a whitelist of headers that you specify. CloudFront caches your objects based on the values in all of the specified headers. CloudFront also forwards the headers that it forwards by default, but it caches your objects based only on the headers that you specify.
- Forward only the default headers. In this configuration, CloudFront doesn't cache your objects based on the values in the request headers.

For the current limit on the number of headers that you can whitelist for each cache behavior, see Amazon CloudFront Limits in the *Amazon Web Services General Reference*. To request a higher limit, go to https://aws.amazon.com/support/createCase?type=service_limit_increase&serviceLimitIncreaseType=cloudfront-distributions.

For information about using the CloudFront console to update a distribution so CloudFront forwards headers to the origin, see Listing, Viewing, and Updating CloudFront Distributions (p. 27). For information about using the CloudFront API to update an existing web distribution, see PUT Distribution Config in the *Amazon CloudFront API Reference*.

Topics

- Selecting the Headers on Which You Want CloudFront to Base Caching (p. 79)
- Configuring CloudFront to Respect Cross-Origin Resource Sharing Settings (p. 80)
- Configuring CloudFront to Cache Objects Based on the Device Type (p. 80)
- Configuring CloudFront to Cache Objects Based on the Language of the Viewer (p. 80)
- Configuring CloudFront to Cache Objects Based on the Location of the Viewer (p. 80)
- Configuring CloudFront to Cache Objects Based on the Protocol of the Request (p. 81)
- How Caching Based on Headers Affects Performance (p. 81)
- How the Case of Headers and Header Values Affects Caching (p. 81)
- Headers that CloudFront Returns to the Viewer (p. 81)

Selecting the Headers on Which You Want CloudFront to Base Caching

The headers that you can forward to the origin and that CloudFront bases caching on depend on whether you're using an Amazon S3 bucket or a custom origin.

- **Amazon S3 –** You can configure CloudFront to forward and to cache your objects based only on the `Origin` header. Amazon S3 doesn't return different objects based on the value of headers, but forwarding the `Origin` header allows CloudFront to distribute content for websites that are enabled for cross-origin resource sharing (CORS). For more information, see Configuring CloudFront to Respect Cross-Origin Resource Sharing Settings (p. 80).
- **Custom origin** – You can configure CloudFront to cache based on the value of any request header except the following:
  - `Accept-Encoding`
  - `Connection`
  - `Cookie` – If you want to forward and cache based on cookies, you use a separate setting in your distribution. For more information, see Configuring CloudFront to Cache Objects Based on Cookies (p. 76).
  - `Proxy-Authorization`
  - `TE`

- `Upgrade`

  You can configure CloudFront to cache objects based on values in the `Date` and `User-Agent` headers, but we don't recommend it. These headers have a lot of possible values, and caching based on their values would cause CloudFront to forward significantly more requests to your origin.

For a full list of HTTP request headers and how CloudFront processes them, see the applicable topic:

- **Amazon S3 origin** – HTTP Request Headers that CloudFront Removes or Updates (p. 103)
- **Custom origin** – HTTP Request Headers and CloudFront Behavior (p. 109)

Configuring CloudFront to Respect Cross-Origin Resource Sharing Settings

If you enabled cross-origin resource sharing (CORS) on an Amazon S3 bucket or a custom origin, you can configure CloudFront to respect the CORS settings. Configure CloudFront to forward a whitelist of headers and include the `Origin` header in the list of headers to forward. For more information about CORS and Amazon S3, see Enabling Cross-Origin Resource Sharing in the *Amazon Simple Storage Service Developer Guide*.

Configuring CloudFront to Cache Objects Based on the Device Type

If you want CloudFront to cache different versions of your objects based on the device a user is using to view your content, configure CloudFront to forward the applicable headers to your custom origin:

- `CloudFront-Is-Mobile-Viewer`
- `CloudFront-Is-Tablet-Viewer`
- `CloudFront-Is-Desktop-Viewer`

Based on the value of the `User-Agent` header, CloudFront sets the value of these headers to `true` or `false` before forwarding the request to your origin. If a device falls into more than one category, more than one value might be `true`. For example, for some tablet devices, CloudFront might set both `CloudFront-Is-Mobile-Viewer` and `CloudFront-Is-Tablet-Viewer` to `true`.

Configuring CloudFront to Cache Objects Based on the Language of the Viewer

If you want CloudFront to cache different versions of your objects based on the language specified in the request, program your application to include the language in the `Accept-Language` header, and configure CloudFront to forward the `Accept-Language` header to your origin.

Configuring CloudFront to Cache Objects Based on the Location of the Viewer

If you want CloudFront to cache different versions of your objects based on the country that the request came from, configure CloudFront to forward the `CloudFront-Viewer-Country` header to your origin. CloudFront automatically converts the IP address that the request came from into a two-letter country code. For an easy-to-use list of country codes, sortable by code and by country name, see the Wikipedia entry ISO 3166-1 alpha-2.

Configuring CloudFront to Cache Objects Based on the Protocol of the Request

If you want CloudFront to cache different versions of your objects based on the protocol of the request, HTTP or HTTPS, configure CloudFront to forward the `CloudFront-Forwarded-Proto` header to your origin.

How Caching Based on Headers Affects Performance

When you configure CloudFront to cache based on one or more headers and the headers have more than one possible value, CloudFront forwards more requests to your origin server for the same object. This slows performance and increases the load on your origin server. If your origin server returns the same object regardless of the value of a given header, we recommend that you don't configure CloudFront to cache based on that header.

If you configure CloudFront to forward more than one header, the order of the headers in viewer requests doesn't affect caching as long as the values are the same. For example, if one request contains the headers A:1,B:2 and another request contains B:2,A:1, CloudFront caches just one copy of the object.

How the Case of Headers and Header Values Affects Caching

When CloudFront caches based on header values, it doesn't consider the case of the header name, but it does consider the case of the header value:

- If viewer requests include both `Product:Acme` and `product:Acme`, CloudFront caches an object only once. The only difference between them is the case of the header name, which doesn't affect caching.
- If viewer requests include both `Product:Acme` and `Product:acme`, CloudFront caches an object twice, because the value is `Acme` in some requests and `acme` in others.

Headers that CloudFront Returns to the Viewer

Configuring CloudFront to forward and cache headers does not affect which headers CloudFront returns to the viewer. CloudFront returns all of the headers that it gets from the origin with a few exceptions. For more information, see the applicable topic:

- **Amazon S3 origins –** See HTTP Response Headers that CloudFront Removes or Updates (p. 105).
- **Custom origins –** See HTTP Response Headers that CloudFront Removes or Updates (p. 114).

Headers and RTMP Distributions

For RTMP distributions, you cannot configure CloudFront to cache your content based on the headers in viewer requests.

Adding, Removing, or Replacing Objects in a Distribution

For information about adding objects to a distribution, see Adding Objects that You Want CloudFront to Distribute (p. 82).

When you replace objects in your distribution, we recommend that you use versioned object names. For more information, see Updating Existing Objects Using Versioned Object Names (p. 82). You can also replace objects with objects that have the same name. See Updating Existing Objects Using the Same Object Names (p. 83). Regardless of how you choose to replace objects in your distribution, we recommend that you specify when objects should be removed from the CloudFront cache. For more information, see Specifying How Long Objects Stay in a CloudFront Edge Cache (Expiration) (p. 83).

If you need to quickly remove objects from a distribution, you can invalidate them. For more information, see Invalidating Objects (Web Distributions Only) (p. 87).

Adding Objects that You Want CloudFront to Distribute

When you want CloudFront to start distributing additional objects, you add the objects to one of the origins that you specified for the distribution, and you expose a CloudFront link to the objects. A CloudFront edge location doesn't fetch the new objects from an origin until the edge location receives viewer requests for the objects. For more information, see How CloudFront Delivers Content (p. 4).

When you add an object that you want CloudFront to distribute, ensure that you add it to one of the Amazon S3 buckets specified in your distribution or, for a custom origin, to a directory in the specified domain. In addition, confirm that the path pattern in the applicable cache behavior sends requests to the correct origin. For example, suppose the path pattern for a cache behavior is `*.html`. If no other cache behaviors are configured to forward requests to that origin, CloudFront will never distribute .jpg files that you upload to the origin.

CloudFront servers don't determine the MIME type for the objects they serve. When you upload an object to your origin, you should set the `Content-Type` header field for the object.

Updating Existing Objects Using Versioned Object Names

When you update existing objects in a CloudFront distribution, we recommend that you include some sort of version identifier either in your object names or in your directory names to give yourself better control over your content. This identifier might be a date-time stamp, a sequential number, or some other method of distinguishing two versions of the same object.

For example, instead of naming a graphic file image.jpg, you might call it image_1.jpg. When you want to start serving a new version of the file, you'd name the new file image_2.jpg, and you'd update your links to point to image_2.jpg. Alternatively, you might put all graphics in an images_v1 directory and, when you want to start serving new versions of one or more graphics, you'd create a new images_v2 directory, and you'd update your links to point to that directory. With versioning, you don't have to wait for an object to expire before CloudFront begins to serve a new version of it, and you don't have to pay for object invalidation.

Even if you version your objects, we still recommend that you set an expiration date. For more information, see Specifying How Long Objects Stay in a CloudFront Edge Cache (Expiration) (p. 83).

> **Note**
> Specifying versioned object names or directory names is not related to Amazon S3 object versioning.

Updating Existing Objects Using the Same Object Names

Although you can update existing objects in a CloudFront distribution and use the same object names, we don't recommend it. CloudFront distributes objects to edge locations only when the objects are requested, not when you put new or updated objects in your origin. If you update an existing object in your origin with a newer version that has the same name, an edge location won't get that new version from your origin until both of the following occur:

- The old version of the object in the cache expires. For more information, see Specifying How Long Objects Stay in a CloudFront Edge Cache (Expiration) (p. 83).
- There's an end user request for the object at that edge location.

If you use the same names when you replace objects, you can't control when CloudFront starts to serve the new files. By default, CloudFront caches objects in edge locations for 24 hours. (For more information, see Specifying How Long Objects Stay in a CloudFront Edge Cache (Expiration) (p. 83).) For example, if you're replacing all of the objects on an entire website:

- Objects for the less popular pages may not be in any edge locations. The new versions of these objects will start being served on the next request.
- Objects for some pages may be in some edge locations and not in others, so your end users will see different versions depending on which edge location they're served from.
- New versions of the objects for the most popular pages might not be served for up to 24 hours because CloudFront might have retrieved the objects for those pages just before you replaced the objects with new versions.

Specifying How Long Objects Stay in a CloudFront Edge Cache (Expiration)

Topics

- Specifying the Minimum Time that CloudFront Caches Objects for Web Distributions (p. 85)
- Specifying the Minimum Time that CloudFront Caches Objects for RTMP Distributions (p. 86)
- Adding Headers to Your Objects Using the Amazon S3 Console (p. 86)

You can control how long your objects stay in a CloudFront cache before CloudFront forwards another request to your origin. Reducing the duration allows you to serve dynamic content. Increasing the duration means your customers get better performance because your objects are more likely to be served directly from the edge cache. A longer duration also reduces the load on your origin.

> **Note**
> You can also control how long errors (for example, 404, not found) stay in a CloudFront cache before CloudFront tries again to get the requested object by forwarding another request to your origin. For more information, see How CloudFront Processes and Caches HTTP 4xx and 5xx Status Codes (p. 115).

Typically, CloudFront serves an object from an edge location until the object expires. After it expires, the next time the edge location gets an end-user request for the object, CloudFront forwards the request to the origin server to verify that the cache contains the latest version of the object:

- If CloudFront already has the latest version, the origin returns only a 304 status code (not modified).

- If CloudFront does not have the latest version, the origin returns a 200 status code (OK) and the latest version of the object.

If an object in an edge location isn't frequently requested, CloudFront might evict the object—remove the object before its expiration date—to make room for objects that are more popular.

By default, each object automatically expires after 24 hours. To specify a different expiration time, configure your origin to add a value for either the `Cache-Control max-age` directive or the `Expires` header field to each object:

- The `Cache-Control max-age` directive lets you specify how long (in seconds) you want the object to remain in the cache before CloudFront gets the object again from the origin server. The minimum expiration time CloudFront supports is 0 seconds for web distributions and 3600 seconds for RTMP distributions. The maximum is in the year 2038. Specify the value in the following format:

  ```
  Cache-Control: max-age=seconds
  ```

  For example, the following directive tells CloudFront to keep the associated object in the cache for 3600 seconds (one hour):

  ```
  Cache-Control: max-age=3600
  ```

  If you want objects to stay in CloudFront edge caches for a different duration than they stay in browser caches, you can use the `Cache-Control max-age` and `Cache-Control s-maxage` directives together. For more information, see Specifying the Minimum Time that CloudFront Caches Objects for Web Distributions (p. 85).
- The `Expires` header field lets you specify an expiration date and time using the format specified in RFC 2616, Hypertext Transfer Protocol -- HTTP/1.1 Section 3.3.1, Full Date, for example:

  ```
  Sat, 30 Jun 2012 23:59:59 GMT
  ```

  Important
  After the expiration date and time in the `Expires` header passes, CloudFront gets the object again from the origin server every time an edge location receives a request for the object.

We recommend that you use the `Cache-Control max-age` directive instead of the `Expires` header field to control object caching. If you specify values both for `Cache-Control max-age` and for `Expires`, CloudFront uses only the value of `max-age`.

You cannot use the HTTP `Cache-Control` or `Pragma` header fields in a `GET` request from an end user to force CloudFront to go back to the origin server for the object. CloudFront ignores those header fields from the end user.

For more information about the `Cache-Control` and `Expires` header fields, see the following sections in *RFC 2616, Hypertext Transfer Protocol -- HTTP/1.1*:

- Section 14.9 Cache Control
- Section 14.21 Expires

For an example of how to add `Cache-Control` and `Expires` header fields using the AWS SDK for PHP, see Upload an Object Using the AWS SDK for PHP in the *Amazon Simple Storage Service Developer Guide*. Some third-party tools are also able to add these fields.

Specifying the Minimum Time that CloudFront Caches Objects for Web Distributions

For web distributions, if you add `Cache-Control` or `Expires` headers to your objects, you can also specify the minimum amount of time that CloudFront keeps an object in the cache before forwarding another request to the origin. These headers also affect how long a browser keeps an object in the cache before forwarding another request to CloudFront.

> **Important**
> If you configure CloudFront to forward all headers to your origin, CloudFront doesn't cache the objects associated with this cache behavior. Instead, it sends every request to the origin. For more information, see Configuring CloudFront to Cache Objects Based on Request Headers (p. 78).

	Minimum TTL = 0 (Default)	**Minimum TTL > 0**
Origin adds `Cache-Control max-age` directive to objects	CloudFront and browsers cache objects for the value of the `Cache-Control max-age` directive.	• **CloudFront caching**—Objects are cached for the greater of the value of the `Cache-Control max-age` directive or the value of the CloudFront **Minimum TTL**. • **Browser caching**—Objects are cached for the value of the `Cache-Control max-age` directive.
Origin does not add `Cache-Control max-age` directive	• **CloudFront caching**—Objects are cached for 24 hours. • **Browser caching**—Depends on the browser.	• **CloudFront caching**—Objects are cached for the greater of 24 hours or the CloudFront **Minimum TTL**. • **Browser caching**— Depends on the browser.
Origin adds `Cache-Control max-age` and `Cache-Control s-maxage` directives to objects	• **CloudFront caching**—Objects are cached for the value of the `Cache-Control s-maxage` directive. • **Browser caching**—Objects are cached for the value of the `Cache-Control max-age` directive.	• **CloudFront caching**—Objects are cached for the greater of the value of the `Cache-Control s-maxage` directive or the value of the CloudFront **Minimum TTL**. • **Browser caching**—Objects are cached for the value of the `Cache-Control max-age` directive.

	Minimum TTL = 0 (Default)	Minimum TTL > 0
Origin adds `Expires` header	CloudFront and browsers cache objects until the date in the `Expires` header. After the date passes, CloudFront forwards every request to the origin.	• **CloudFront caching**—Objects are cached until the date in the `Expires` header. After the date passes, objects are cached for the value of the CloudFront **Minimum TTL**. • **Browser caching**—Objects are cached until the date in the `Expires` header. After the date passes, browsers forward every request to CloudFront.
Origin adds `Cache-Control` `no-cache`, `no-store`, and/or `private` directives to objects	CloudFront and browsers respect the headers.	• **CloudFront caching**—Objects are cached for the **Minimum TTL**. • **Browser caching**—Browsers respect the headers.

For information about how to change settings for web distributions using the CloudFront console, see Listing, Viewing, and Updating CloudFront Distributions (p. 27). For information about how to change settings for web distributions using the CloudFront API, see PUT Config.

Specifying the Minimum Time that CloudFront Caches Objects for RTMP Distributions

For RTMP distributions, CloudFront keeps objects in edge caches for 24 hours by default. You can add `Cache-Control` or `Expires` headers to your objects to reduce the cache duration to as little as an hour (3600 seconds) before CloudFront forwards another request to the origin. If you specify a lower value, CloudFront uses 3600 seconds.

Adding Headers to Your Objects Using the Amazon S3 Console

> **Note**
> Using the Amazon S3 console, you can only add headers to one object at a time, but with some third-party tools, you can add headers to multiple Amazon S3 objects at a time. For more information about third-party tools that support Amazon S3, perform a web search on **`AWS S3 third party tools`**.

To add a `Cache-Control` or `Expires` header field to Amazon S3 objects using the Amazon S3 console

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In the Amazon S3 console, in the Buckets pane, click the name of the bucket that contains the files.
3. In the **Objects and Folders** pane, select the first object to which you want to add a header field.
4. At the top of the **Objects and Folders** pane, click **Actions** and click **Properties**.
5. In the **Properties** pane, click the **Metadata** tab.

6. On the Metadata tab, click **Add More Metadata**.
7. In the **Key** list, click **Cache-Control** or **Expires**, as applicable.
8. In the **Value** field, enter the applicable value:

   - For a `Cache-Control` field, enter:

     ```
     max-age=number of seconds that you want objects to stay in a CloudFront
     edge cache
     ```
   - For an **Expires** field, enter a date and time in HTML format.

9. Click **Save**.
10. If you want to add a header field to additional objects, in the **Objects and Folders** pane, click the name of the next object, and repeat Steps 6 through 9.

Invalidating Objects (Web Distributions Only)

If you need to remove an object from CloudFront edge-server caches before it expires, you can do one of the following:

- Invalidate the object. The next time a viewer requests the object, CloudFront returns to the origin to fetch the latest version of the object.
- Use object versioning to serve a different version of the object that has a different name. For more information, see Updating Existing Objects Using Versioned Object Names (p. 82).

> **Important**
> You can invalidate most types of objects that are served by a web distribution, but you cannot invalidate media files in the Microsoft Smooth Streaming format when you have enabled Smooth Streaming for the corresponding cache behavior. In addition, you cannot invalidate objects that are served by an RTMP distribution.

You can invalidate a specified number of objects each month for free. Above that limit, you pay a fee for each object that you invalidate. For example, to invalidate a directory and all of the files in the directory, you must invalidate the directory and each file individually. If you need to invalidate a lot of files, it might be easier and less expensive to create a new distribution and change your object paths to refer to the new distribution. For more information about the charges for invalidation, see Paying for Object Invalidation (p. 91).

Topics

- Choosing Between Invalidating Objects and Using Versioned Object Names (p. 87)
- Determining Which Objects to Invalidate (p. 88)
- Invalidating Objects and Displaying Information about Invalidations (p. 88)
- Third-Party Tools for Invalidating Objects (p. 91)
- Invalidation Limits (p. 91)
- Paying for Object Invalidation (p. 91)

Choosing Between Invalidating Objects and Using Versioned Object Names

To control the versions of objects served from your distribution, you can either invalidate objects or give them versioned file names. If you'll want to update your objects frequently, we recommend that you primarily use object versioning for the following reasons:

- Versioning enables you to control which object a request returns even when the end user has a version cached either locally or behind a corporate caching proxy. If you invalidate the object, the end user may continue to see the old version until it expires from those caches.
- File names are included in the CloudFront access logs, so versioning makes it easier to analyze the results of object changes.
- Versioning provides a way to serve different versions of objects to different end users.
- Versioning simplifies rolling forward and back between object revisions.
- Versioning is less expensive. You still have to pay for CloudFront to transfer new versions of your objects to edge locations, but you don't have to pay the per-file charge for invalidating objects.

For more information about object versioning, see Updating Existing Objects Using Versioned Object Names (p. 82).

Determining Which Objects to Invalidate

If you want to invalidate all of your objects in all CloudFront edge caches but your users don't necessarily access every object on your origin, you can determine which objects viewers have requested from CloudFront and invalidate only those objects. To determine which objects viewers have requested, enable CloudFront access logging. For more information about access logs, see Access Logs (p. 182).

Invalidating Objects and Displaying Information about Invalidations

You can use the CloudFront console or CloudFront API actions to create and run an invalidation, display a list of the invalidations that you submitted previously, and display detailed information about an individual invalidation. You can also copy an existing invalidation, edit the list of object paths, and run the edited invalidation.

See the applicable topic:

- Invalidating Objects Using the CloudFront Console (p. 88)
- Copying, Editing, and Rerunning an Existing Invalidation Using the CloudFront Console (p. 90)
- Listing Invalidations Using the CloudFront Console (p. 90)
- Displaying Information about an Invalidation Using the CloudFront Console (p. 91)
- Invalidating Objects and Displaying Information about Invalidations Using the CloudFront API (p. 91)

Invalidating Objects Using the CloudFront Console

You can create any number of invalidations, but you can have only three invalidations per distribution in progress at one time. To determine how many invalidations are currently in progress, see the **Status** column on the **Invalidations** tab.

Note the following about specifying the objects that you want to invalidate:

Case sensitivity
Object paths are case sensitive, so `images/image.jpg` and `images/Image.jpg` specify two different objects.

Default root object
To invalidate the default root object, specify the path the same way you specify the path for any other object.

Distribution types
You can only invalidate objects that are associated with a web distribution.

Forwarding cookies

If you configured CloudFront to forward cookies to your origin, CloudFront edge caches might contain several versions of the object. When you invalidate an object, CloudFront invalidates every cached version of the object regardless of its associated cookies. You can't selectively invalidate some versions and not others based on the associated cookies. For more information, see Configuring CloudFront to Cache Objects Based on Cookies (p. 76).

Forwarding headers

If you configured CloudFront to forward a whitelist of headers to your origin and to cache based on the values of the headers, CloudFront edge caches might contain several versions of the object. When you invalidate an object, CloudFront invalidates every cached version of the object regardless of the header values. You can't selectively invalidate some versions and not others based on header values. (If you configure CloudFront to forward all headers to your origin, CloudFront doesn't cache your objects.) For more information, see Configuring CloudFront to Cache Objects Based on Request Headers (p. 78).

Forwarding query strings

If you configured CloudFront to forward query strings to your origin, you must include the query strings when invalidating objects, for example:

- `images/image.jpg?parameter1=a`
- `images/image.jpg?parameter1=b`

If client requests include five different query strings for the same object, you must invalidate that object five times, once for each query string. For more information, see Configuring CloudFront to Cache Based on Query String Parameters (p. 75). To determine which query strings are in use, you can enable CloudFront logging. For more information, see Access Logs (p. 182).

Non-ASCII or unsafe characters in the path

If the path includes non-ASCII characters or unsafe characters as defined in RFC 1783 (http://www.ietf.org/rfc/rfc1738.txt), URL-encode those characters. Do not URL-encode any other characters in the path, or CloudFront will not invalidate the old version of the updated object.

Number of objects

You can specify up to 1,000 objects in one invalidation.

Object paths

The path is relative to the distribution. A leading / is optional. For example, to invalidate the object at `http://d111111abcdef8.cloudfront.net/images/image2.jpg`, you would specify:

```
/images/image2.jpg
```

or

```
images/image2.jpg
```

You must explicitly invalidate every object and every directory that you want CloudFront to stop serving. You cannot use wildcards to invalidate groups of objects, and you cannot invalidate all of the objects in a directory by specifying the directory path.

If the object is a directory and if you have not standardized on a method for specifying directories—with or without a trailing slash (/)—we recommend that you invalidate the directory both with and without a trailing slash, for example, `images` and `images/`. For more information, see How Public URLs Affect the Invalidation of Directories (p. 74).

The maximum length of a path is 4,000 characters.

Signed URLs

If you are using signed URLs, invalidate the object by including only the portion of the URL before the question mark (?).

To invalidate objects using the CloudFront console

1. Sign in to the AWS Management Console and open the Amazon CloudFront console at https://console.aws.amazon.com/cloudfront/.
2. Click the distribution for which you want to invalidate objects.
3. Click **Distribution Settings**.
4. Click the **Invalidations** tab.
5. Click **Create Invalidation**.
6. Enter the paths of the objects that you want to invalidate.
7. Click **Invalidate**.

Copying, Editing, and Rerunning an Existing Invalidation Using the CloudFront Console

You can copy an invalidation that you created previously, update the list of object paths, and run the updated validation. You cannot copy an existing invalidation, update the object paths, and save the updated invalidation without running it.

> **Important**
> If you copy an invalidation that is still in progress, update the list of object paths, and run the updated invalidation, CloudFront will not stop or delete the invalidation that you copied. If any object paths appear in the original and in the copy, CloudFront will try to invalidate the objects twice, and both invalidations will count against your maximum number of free invalidations for the month. If you've already reached the maximum number of free validations, you'll be charged for both invalidations of each object. For more information, see Invalidation Limits (p. 91).

To copy, edit, and rerun an existing invalidation using the CloudFront console

1. Sign in to the AWS Management Console and open the Amazon CloudFront console at https://console.aws.amazon.com/cloudfront/.
2. Click the distribution that contains the invalidation that you want to copy.
3. Click **Distribution Settings**.
4. Click the **Invalidations** tab.
5. Click the invalidation that you want to copy.

   If you aren't sure which invalidation you want to copy, you can click an invalidation and click **Details** to display detailed information about that invalidation.
6. Click **Copy**.
7. Update the list of object paths if applicable.
8. Click **Invalidate**.

Listing Invalidations Using the CloudFront Console

Using the console, you can display a list of the last 100 invalidations that you've created and run for a distribution. If you want to get a list of more than 100 invalidations, use the GET Invalidation List API action. For more information, go to GET Invalidation List in the *Amazon CloudFront API Reference*.

To list invalidations using the CloudFront console

1. Sign in to the AWS Management Console and open the Amazon CloudFront console at https://console.aws.amazon.com/cloudfront/.
2. Click the distribution for which you want to display a list of invalidations.
3. Click **Distribution Settings**.

4. Click the **Invalidations** tab.

Displaying Information about an Invalidation Using the CloudFront Console

You can display detailed information about an invalidation, including distribution ID, invalidation ID, the status of the invalidation, the date and time that the invalidation was created, and a complete list of the object paths.

To display information about an invalidation using the CloudFront console

1. Sign in to the AWS Management Console and open the Amazon CloudFront console at https://console.aws.amazon.com/cloudfront/.
2. Click the distribution that contains the invalidation about which you want to display detailed information.
3. Click **Distribution Settings**.
4. Click the **Invalidations** tab.
5. Click the invalidation about which you want to display detailed information.
6. Click **Details**.

Invalidating Objects and Displaying Information about Invalidations Using the CloudFront API

For information about invalidating objects and about displaying information about invalidations using the CloudFront API, see the applicable topic in the *Amazon CloudFront API Reference*:

- Invalidating objects: POST Invalidation
- Getting a list of your invalidations: GET Invalidation List
- Getting information about a specific invalidation: GET Invalidation

Third-Party Tools for Invalidating Objects

In addition to the invalidation methods provided by CloudFront, several third-party tools provide ways to invalidate objects. For a list of tools, see Invalidating Objects (p. 285).

Invalidation Limits

You can make any number of invalidation requests, but you can have only three invalidation requests per distribution in progress at one time. Each request can contain up to 1,000 objects to invalidate. If you exceed these limits, CloudFront returns an error message.

> **Note**
> It usually takes 10 to 15 minutes for CloudFront to complete your invalidation request, depending on the size of the request.

Paying for Object Invalidation

The first 1,000 object invalidations you request per month are free; you pay for each object invalidation over 1,000 in a month. This limit applies to the total number of object invalidations across all of the distributions that you create with one AWS account. For example, if you use the AWS account `john@example.com` to create three distributions, and each distribution has 600 object invalidations in a given month (for a total of 1,800 invalidations), AWS will charge you for 800 object invalidations in that month. For specific information about invalidation pricing, go to Amazon CloudFront Pricing.

Note

For the purposes of invalidation pricing, an object invalidation request is defined as a single `Path` element object. For more information about the `Path` element, see Invalidating Objects and Displaying Information about Invalidations (p. 88).

Customizing Error Responses

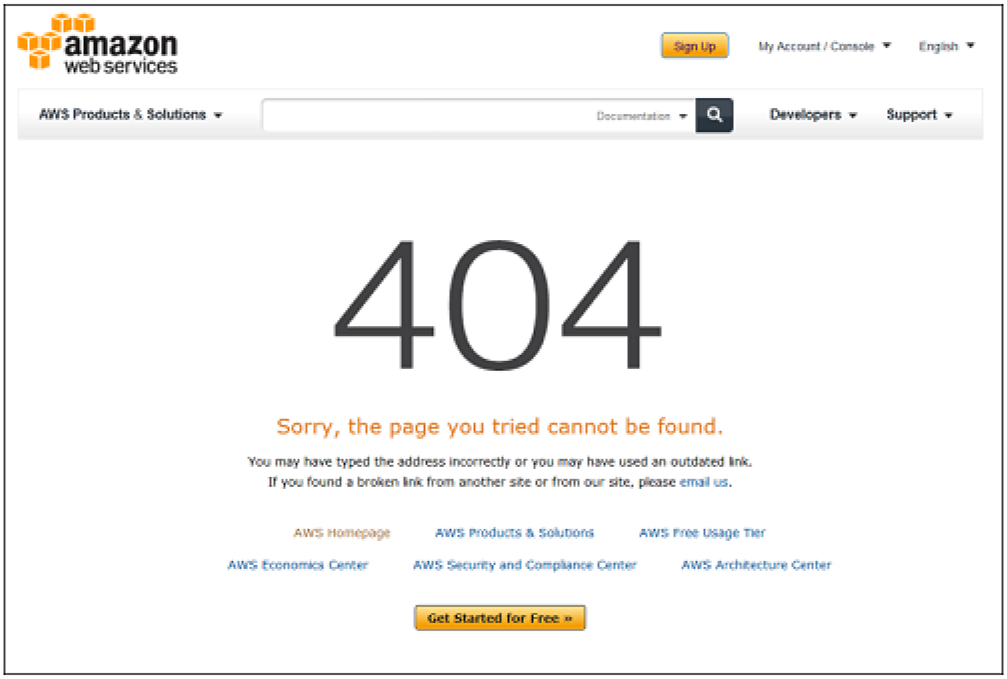
Topics

- Creating or Updating a Cache Behavior for Custom Error Pages (p. 93)
- Changing Response Codes (p. 93)
- Controlling How Long CloudFront Caches Errors (p. 94)
- How CloudFront Responds When a Custom Error Page Is Unavailable (p. 94)
- Pricing for Custom Error Pages (p. 94)
- Configuring Error Response Behavior (p. 95)

If the objects that you're serving through CloudFront are unavailable for some reason, your web server typically returns an HTTP status code to CloudFront. For example, if a viewer specifies an invalid URL, your web server returns a 404 status code to CloudFront, and CloudFront returns that status code to the viewer. The viewer displays a brief and sparsely formatted default message similar to this:

```
Not Found: The requested URL /myfilename.html was not found on this server.
```

If you'd rather display a custom error message, possibly using the same formatting as the rest of your website, you can have CloudFront return to the viewer an object (for example, an HTML file) that contains your custom error message.



404

Sorry, the page you tried cannot be found.

You may have typed the address incorrectly or you may have used an outdated link.
If you found a broken link from another site or from our site, please email us.

You can specify a different object for each supported HTTP status code, or you can use the same object for all of the supported status codes. You can also choose to specify objects for some status codes and not for others.

The objects that you're serving through CloudFront can be unavailable for a variety of reasons. These fall into two broad categories:

- **Client errors** indicate a problem with the request. For example, an object with the specified name isn't available, or the user doesn't have the permissions required to get an object in your Amazon S3 bucket. When a client error occurs, the origin returns an HTTP status code in the 400 range to CloudFront.
- **Server errors** indicate a problem with the origin server. For example, the HTTP server is busy or unavailable. When a server error occurs, either your origin server returns an HTTP status code in the 500 range to CloudFront, or CloudFront doesn't get a response from your origin server for a certain period of time and assumes a 504 status code (gateway timeout).

The HTTP status codes for which CloudFront can return a custom error page include the following:

- 400, 403, 404, 405, 414
- 500, 501, 502, 503, 504

For a detailed explanation of how CloudFront handles error responses from your origin, see How CloudFront Processes and Caches HTTP 4xx and 5xx Status Codes (p. 115).

Creating or Updating a Cache Behavior for Custom Error Pages

If you want to store your objects and your custom error pages in different locations, your distribution must include a cache behavior for which the following is true:

- The value of **Path Pattern** matches the path to your custom error messages. For example, suppose you saved custom error pages for 4xx errors in an Amazon S3 bucket in a directory named `/4xx-errors`. Your distribution must include a cache behavior for which the path pattern routes requests for your custom error pages to that location, for example, **/4xx-errors/***.
- The value of **Origin** specifies the value of **Origin ID** for the origin that contains your custom error pages.

For more information, see Cache Behavior Settings (p. 44) in the topic Values that You Specify When You Create or Update a Web Distribution (p. 40).

Changing Response Codes

You can choose the HTTP status code CloudFront returns along with a custom error page for a given HTTP status code. For example, if your origin returns a 500 status code to CloudFront, you might want CloudFront to return a custom error page and a 200 status code (OK) to the viewer. There are a variety of reasons that you might want CloudFront to return a status code different from the one that your origin returned:

- Some Internet devices (some firewalls and corporate proxies, for example) intercept HTTP 4xx and 5xx and prevent the response from being returned to the viewer. If you substitute `200`, the response typically won't be intercepted.
- If you don't care about distinguishing among different client errors or server errors, you can specify **400** or **500** as the value that CloudFront returns for all 4xx or 5xx status codes.
- You might want to return a `200` status code (OK) and static website so your customers don't know that your website is down.

HTTP status codes that CloudFront can return along with a custom error page include the following:

- 200
- 400, 403, 404, 405, 414
- 500, 501, 502, 503, 504

Controlling How Long CloudFront Caches Errors

By default, when your origin returns an HTTP 4xx or 5xx status code, CloudFront caches these error responses for five minutes and then submits the next request for the object to your origin to see whether the problem that caused the error has been resolved and the requested object is now available.

You can specify (p. 95) the error-caching duration—the **Error Caching Minimum TTL**—for each 4xx and 5xx status code that CloudFront caches. When you specify a duration, note the following:

- If you specify a short error-caching duration, CloudFront forwards more requests to your origin than if you specify a longer duration. For 5xx errors, this may aggravate the problem that originally caused your origin to return an error.
- When your origin returns an error for an object, CloudFront responds to requests for the object either with the error response or with your custom error page until the error-caching duration elapses. If you specify a long error-caching duration, CloudFront might continue to respond to requests with an error response or your custom error page for a long time after the object becomes available again.

If you want to control how long CloudFront caches errors for individual objects, you can configure your origin server to add the applicable header to the error response for that object:

- **If the origin adds a `Cache-Control max-age` or `Cache-Control s-maxage` directive, or an `Expires` header:** CloudFront caches error responses for the greater of the value in the header or the value of **Error Caching Minimum TTL**.
- **If the origin adds other `Cache-Control` directives or adds no headers:** CloudFront caches error responses for the value of **Error Caching Minimum TTL**.

If the expiration time for a 4xx or 5xx status code for an object is longer than you want to wait, you can invalidate the status code by using the URL of the requested object. If your origin is returning an error response for multiple objects, you need to invalidate each object separately. For more information about invalidating objects, see Invalidating Objects (Web Distributions Only) (p. 87).

How CloudFront Responds When a Custom Error Page Is Unavailable

If you configure CloudFront to return a custom error page for an HTTP status code but the custom error page isn't available, CloudFront returns to the viewer the status code that CloudFront received from the origin that contains the custom error pages. For example, suppose your custom origin returns a 500 status code and you have configured CloudFront to get a custom error page for a 500 status code from an Amazon S3 bucket. However, someone accidentally deleted the custom error page from your bucket. CloudFront will return an HTTP 404 status code (not found) to the viewer that requested the object.

Pricing for Custom Error Pages

When CloudFront returns a custom error page to a viewer, you pay the standard CloudFront charges for the custom error page, not the charges for the requested object. For more information about CloudFront charges, see CloudFront Billing and Usage Reports (p. 7).

Configuring Error Response Behavior

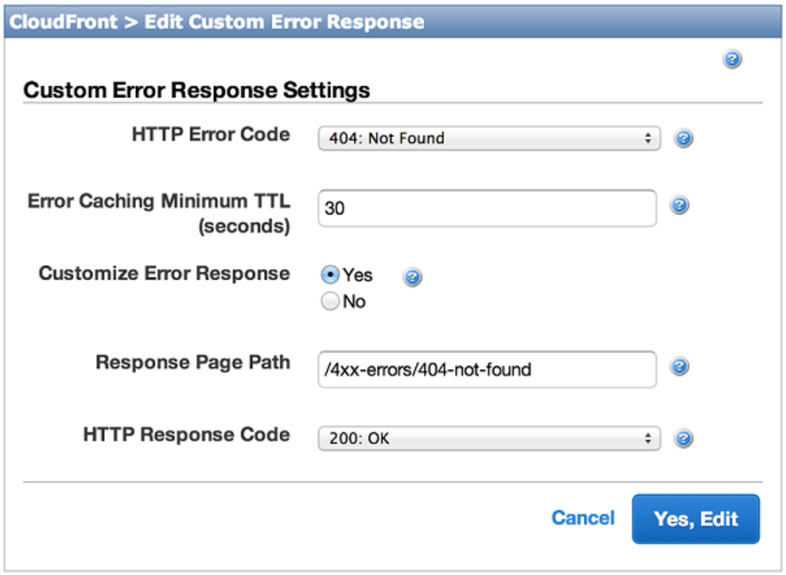
You can use either the CloudFront API or console to configure CloudFront error responses. For information about using the CloudFront API to configure error responses, go to PUT Distribution Config in the *Amazon CloudFront API Reference*, and see the `CustomErrorResponses` element.

To configure CloudFront error responses using the console

1. Create the custom error pages that you want CloudFront to return to viewers when your origin returns HTTP 4xx or 5xx errors. Save the pages in a location that is accessible to CloudFront.

   We recommend that you store custom error pages in an Amazon S3 bucket even if you're using a custom origin. If you store custom error pages on an HTTP server and the server starts to return 5xx errors, CloudFront can't get the files that you want to return to viewers because the origin server is unavailable.
2. Confirm that you have granted CloudFront at least `read` permission to your custom error page objects.

   For more information about Amazon S3 permissions, see Access Control in the *Amazon Simple Storage Service Developer Guide*. For information on using the Amazon S3 console to update permissions, go to the *Amazon Simple Storage Service Console User Guide*.
3. (Optional) Configure your origin server to add `Cache-Control` directives or an `Expires` header along with the error response for specific objects, if applicable. For more information, see Controlling How Long CloudFront Caches Errors (p. 94).
4. Sign in to the AWS Management Console and open the Amazon CloudFront console at https://console.aws.amazon.com/cloudfront/.
5. In the list of distributions, select the distribution to update and click **Distribution Settings**.
6. Click the **Error Pages** tab. Then either click **Create Custom Error Response**, or select an existing error code and click **Edit**.



Custom Error Response Settings

HTTP Error Code: 404: Not Found

Error Caching Minimum TTL (seconds): 30

Customize Error Response: ● Yes ○ No

Response Page Path: /4xx-errors/404-not-found

HTTP Response Code: 200: OK

7. Enter the applicable values. For more information, see Custom Error Pages and Error Caching (p. 53).
8. If you configured CloudFront to return custom error pages, add or update the applicable cache behaviors. For more information, see Creating or Updating a Cache Behavior for Custom Error Pages (p. 93).
9. To save your changes, click **Yes, Edit**.

How CloudFront Processes Partial Requests for an Object (Range GETs)

For a large object, an end user's browser or client might make multiple `GET` requests and use the `Range` request header to download the object in smaller units. These requests for ranges of bytes, sometimes known as `Range GET` requests, improve the efficiency of partial downloads and the recovery from partially failed transfers.

When CloudFront receives a `Range GET` request, it checks the cache in the edge location that received the request. If the cache in that edge location already contains the entire object or the requested portion of the object, CloudFront immediately serves the requested range from the cache.

If the cache doesn't contain the requested range, CloudFront forwards the request to the origin. (To optimize performance, CloudFront may request a larger range than the client requested in the `Range GET`.) What happens next depends on whether the origin supports `Range GET` requests:

- **If the origin supports `Range GET` requests:** It returns the requested range. CloudFront serves the requested range and also caches it for future requests. (Amazon S3 supports `Range GET` requests, as do some HTTP servers, for example, Apache and IIS. For information about whether your HTTP server does, see the documentation for your HTTP server.)
- **If the origin doesn't support `Range GET` requests:** It returns the entire object. CloudFront serves the entire object and also caches the entire object for future requests. After CloudFront caches the entire object in an edge cache, it responds to `Range GET` requests by serving the requested range.

In either case, CloudFront begins to serve the requested range or object to the end user as soon as the first byte arrives from the origin.

CloudFront generally follows the RFC specification for the `Range` header. However, if your `Range` headers don't adhere to the following requirements, CloudFront will return HTTP status code 200 with the full object instead of status code 206 with the specified ranges:

- The ranges must be listed in ascending order. For example, `100-200,300-400` is valid, `300-400,100-200` is not valid.
- The ranges must not overlap. For example, `100-200,150-250` is not valid.
- All of the ranges specifications must be valid. For example, you can't specify a negative value as part of a range.

For more information about the `Range` request header, see "Section 14.35 Range" in *Hypertext Transfer Protocol -- HTTP/1.1* at http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35.

Specifying a Default Root Object (Web Distributions Only)

You can configure CloudFront to return a specific object (the default root object) when an end user requests the root URL for your distribution instead of an object in your distribution. Specifying a default root object avoids exposing the contents of your distribution.

For example, the following request points to the object `image.jpg`:

```
http://d111111abcdef8.cloudfront.net/image.jpg
```

The following request points to the root URL of the same distribution instead of to a specific object:

```
http://d111111abcdef8.cloudfront.net/
```

When you define a default root object, an end-user request that calls the root of your distribution returns the default root object. For example, if you designate the file `index.html` as your default root object, a request for:

```
http://d111111abcdef8.cloudfront.net/
```

returns:

```
http://d111111abcdef8.cloudfront.net/index.html
```

However, if you define a default root object, an end-user request for a subdirectory of your distribution does not return the default root object. For example, suppose `index.html` is your default root object and that CloudFront receives an end-user request for the `install` directory under your CloudFront distribution:

```
http://d111111abcdef8.cloudfront.net/install/
```

CloudFront will not return the default root object even if a copy of `index.html` appears in the `install` directory.

If you configure your distribution to allow all of the HTTP methods that CloudFront supports, the default root object applies to all methods. For example, if your default root object is index.php and you write your application to submit a `POST` request to the root of your domain (http://example.com), CloudFront will send the request to http://example.com/index.php.

The behavior of CloudFront default root objects is different from the behavior of Amazon S3 index documents. When you configure an Amazon S3 bucket as a website and specify the index document, Amazon S3 returns the index document even if a user requests a subdirectory in the bucket. (A copy of the index document must appear in every subdirectory.) For more information about configuring Amazon S3 buckets as websites and about index documents, see the Hosting Websites on Amazon S3 chapter in the *Amazon Simple Storage Service Developer Guide*.

> **Important**
> Remember that a default root object applies only to your CloudFront distribution. You still need to manage security for your origin. For example, if you are using an Amazon S3 origin, you still need to set your Amazon S3 bucket ACLs appropriately to ensure the level of access you want on your bucket.

If you don't define a default root object, requests for the root of your distribution pass to your origin server. If you are using an Amazon S3 origin, any of the following might be returned:

- **A list of the contents of your Amazon S3 bucket**—Under any of the following conditions, the contents of your origin are visible to anyone who uses CloudFront to access your distribution:
  - Your bucket is not properly configured.
  - The Amazon S3 permissions on the bucket associated with your distribution and on the objects in the bucket grant access to *everyone*.
  - An end user accesses your origin using your origin root URL.
- **A list of the private contents of your origin**—If you configure your origin as a private distribution (only you and CloudFront have access), the contents of the Amazon S3 bucket associated with your distribution are visible to anyone who has the credentials to access your distribution through CloudFront. In this case, users are not able to access your content through your origin root URL. For more information about distributing private content, see Serving Private Content through CloudFront (p. 118).
- **Error 403 Forbidden**—CloudFront returns this error if the permissions on the Amazon S3 bucket associated with your distribution or the permissions on the objects in that bucket deny access to CloudFront and to everyone.

To avoid exposing the contents of your web distribution or returning an error, perform the following procedure to specify a default root object for your distribution.

To specify a default root object for your distribution

1. Upload the default root object to the origin that your distribution points to.

   The file can be any type supported by CloudFront. For a list of constraints on the file name, see the description of the `DefaultRootObject` element in DistributionConfig Complex Type.

   > **Note**
   > If the file name of the default root object is too long or contains an invalid character, CloudFront returns the error `HTTP 400 Bad Request - InvalidDefaultRootObject`. In addition, CloudFront caches the code for five minutes and writes the results to the access logs.

2. Confirm that the permissions for the object grant CloudFront at least `read` access.

   For more information about Amazon S3 permissions, see Access Control in the *Amazon Simple Storage Service Developer Guide*. For information on using the Amazon S3 console to update permissions, go to the *Amazon Simple Storage Service Console User Guide*.
3. Update your distribution to refer to the default root object using the CloudFront console or the CloudFront API.

   To specify a default root object using the CloudFront console:

   a. Sign in to the AWS Management Console and open the Amazon CloudFront console at https://console.aws.amazon.com/cloudfront/.
   b. In the list of distributions in the top pane, select the distribution to update.
   c. In the **Distribution Details** pane, on the **General** tab, click **Edit**.
   d. In the **Edit Distribution** dialog box, in the **Default Root Object** field, enter the file name of the default root object.

      Enter only the object name, for example, `index.html`. Do not add a `/` before the object name.
   e. To save your changes, click **Yes, Edit**.

   To update your configuration using the CloudFront API, you specify a value for the `DefaultRootObject` element in your distribution. For information about using the CloudFront API to specify a default root object, see PUT Distribution Config in the *Amazon CloudFront API Reference*.
4. Confirm that you have enabled the default root object by requesting your root URL. If your browser doesn't display the default root object, perform the following steps:

   a. Confirm that your distribution is fully deployed by viewing the status of your distribution in the CloudFront console.
   b. Repeat Steps 2 and 3 to verify that you granted the correct permissions and that you correctly updated the configuration of your distribution to specify the default root object.

Serving Compressed Files

Amazon CloudFront can serve both compressed and uncompressed files from an origin server. CloudFront relies on the origin server either to compress the files or to have compressed and uncompressed versions of files available; CloudFront does not perform the compression on behalf of the origin server. With some qualifications, CloudFront can also serve compressed content from Amazon S3. For more information, see Choosing the File Types to Compress (p. 100).

Serving compressed content makes downloads faster because the files are smaller—in some cases, less than half the size of the original. Especially for JavaScript and CSS files, faster downloads translates into faster rendering of web pages for your users. In addition, because the cost of CloudFront data transfer is based on the total amount of data served, serving compressed files is less expensive than serving uncompressed files.

CloudFront can only serve compressed data if the viewer (for example, a web browser or media player) requests compressed content by including `Accept-Encoding: gzip` in the request header. The content must be compressed using gzip; other compression algorithms are not supported. If the request header includes additional content encodings, for example, `deflate` or `sdch`, CloudFront removes them before forwarding the request to the origin server. If `gzip` is missing from the `Accept-Encoding` field, CloudFront serves only the uncompressed version of the file. For more information about the `Accept-Encoding` request-header field, see "Section 14.3 Accept Encoding" in *Hypertext Transfer Protocol -- HTTP/1.1* at http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html.

For more information, see the following topics:

Topics

- How CloudFront Serves Compressed Content from a Custom Origin (p. 99)
- Serving Compressed Files from Amazon S3 (p. 99)
- Choosing the File Types to Compress (p. 100)

How CloudFront Serves Compressed Content from a Custom Origin

Here's how CloudFront commonly serves compressed content from a custom origin to a web application:

1. You configure your web server to compress selected file types. For more information, see Choosing the File Types to Compress (p. 100).
2. You create a CloudFront distribution.
3. You program your web application to access files using CloudFront URLs.
4. A user accesses your application in a web browser.
5. CloudFront directs web requests to the edge location that has the lowest latency for the user, which may or may not be the geographically closest edge location.
6. At the edge location, CloudFront checks the cache for the object referenced in each request. If the browser included `Accept-Encoding: gzip` in the request header, CloudFront checks for a compressed version of the file. If not, CloudFront checks for an uncompressed version.
7. If the file is in the cache, CloudFront returns the file to the web browser. If the file is not in the cache:

    a. CloudFront forwards the request to the origin server.
    b. If the request is for a type of file that you want to serve compressed (see Step 1), the web server compresses the file.
    c. The web server returns the file (compressed or uncompressed, as applicable) to CloudFront.
    d. CloudFront adds the file to the cache and serves the file to the user's browser.

Serving Compressed Files from Amazon S3

If you want to serve compressed files from Amazon S3:

1. Create two versions of each file, one compressed and one uncompressed. To ensure that the compressed and uncompressed versions of a file don't overwrite one another in the CloudFront cache, give each file a unique name, for example, welcome.js and welcome.js.gz.
2. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
3. Upload both versions to Amazon S3.

   If you're using the Amazon S3 API to upload files, include a `Content-Type` header to specify the MIME type of each file. The default MIME type is `binary/octet-stream`.

   If you're using the Amazon S3 console to upload files, Amazon S3 automatically figures out the content types of the files by default.
4. Add a `Content-Encoding` header field for each compressed file and set the field value to `gzip`.

   For an example of how to add a `Content-Encoding` header field using the AWS SDK for PHP, see Upload an Object Using the AWS SDK for PHP in the *Amazon Simple Storage Service Developer Guide*. Some third-party tools are also able to add this field.

   To add a `Content-Encoding` header field and set the field value using the Amazon S3 console, perform the following procedure:

   a. In the Amazon S3 console, in the **Buckets** pane, click the name of the bucket that contains the compressed files.
   b. Click the name of a file for which you want to add a `Content-Encoding` header.
   c. At the top of the page, click **Actions**. In the **Actions** list, click **Properties**.
   d. In the right pane, click **Metadata**.
   e. Under **Metadata**, click **Add More Metadata**.
   f. In the **Key** list, click **Content-Encoding**.
   g. In the **Value** field, enter **gzip**.
   h. Click **Save**.
   i. Repeat steps 4b through 4h for the remaining compressed files.

5. When generating HTML that links to content in CloudFront (for example, using php, asp, or jsp), evaluate whether the request from the viewer includes `Accept-Encoding: gzip` in the request header. If so, rewrite the corresponding link to point to the compressed object name.

Choosing the File Types to Compress

Some types of files compress well, for example, HTML, CSS, and JavaScript files. Some types of files may compress a few percent, but not enough to justify the additional processor cycles required for your web server to compress the content, and some types of files even get larger when they're compressed. File types that generally don't compress well include graphic files that are already compressed (.jpg, .gif), video formats, and audio formats. We recommend that you test compression for the file types in your distribution to ensure that there is sufficient benefit to compression.

Request and Response Behavior

The following sections explain how CloudFront processes viewer requests and forwards the requests to your Amazon S3 or custom origin, and how CloudFront processes responses from your origin, including how CloudFront processes and caches 4xx and 5xx HTTP status codes.

Topics

- Request and Response Behavior for Amazon S3 Origins (p. 101)
- Request and Response Behavior for Custom Origins (p. 106)
- How CloudFront Processes and Caches HTTP 4xx and 5xx Status Codes (p. 115)

Request and Response Behavior for Amazon S3 Origins

Topics

- How CloudFront Processes and Forwards Requests to Your Amazon S3 Origin Server (p. 101)
- How CloudFront Processes Responses from Your Amazon S3 Origin Server (p. 104)

How CloudFront Processes and Forwards Requests to Your Amazon S3 Origin Server

For information about how CloudFront processes viewer requests and forwards the requests to your Amazon S3 origin, see the applicable topic:

Topics

- Caching Duration and Minimum TTL (p. 102)
- Client IP Addresses (p. 102)
- Conditional GETs (p. 102)
- Cookies (p. 102)
- Cross-Origin Resource Sharing (CORS) (p. 103)
- HTTP Methods (p. 103)
- HTTP Request Headers that CloudFront Removes or Updates (p. 103)

- Maximum Length of a Request and Maximum Length of a URL (p. 104)
- Protocols (p. 104)
- Query Strings (p. 104)
- Request Timeout (p. 104)

Caching Duration and Minimum TTL

For web distributions, to control how long your objects stay in a CloudFront cache before CloudFront forwards another request to your origin, you can:

- Configure your origin to add a `Cache-Control` or an `Expires` header field to each object.
- Specify a value for Minimum TTL in CloudFront cache behaviors.
- Use the default value of 24 hours.

For more information, see Specifying How Long Objects Stay in a CloudFront Edge Cache (Expiration) (p. 83).

Client IP Addresses

CloudFront forwards the client IP address to Amazon S3 in the `X-Forwarded-For` request header. The `X-Forwarded-For` request header looks like this:

```
X-Forwarded-For: 192.0.2.235
```

If the `X-Forwarded-For` request header contains two or more IP addresses, the first one is always the client IP address. The other addresses are for each successive proxy that passes along the request; the last IP address is for the proxy that forwarded the request to CloudFront:

```
X-Forwarded-For: client-IP-address, proxy-IP-address, another-proxy-IP-address
```

Conditional GETs

When CloudFront receives a request for an object that has expired from an edge cache, it forwards the request to the Amazon S3 origin either to get the latest version of the object or to get confirmation from Amazon S3 that the CloudFront edge cache already has the latest version. When Amazon S3 originally sent the object to CloudFront, it included an `ETag` value and a `LastModified` value in the response. In the new request that CloudFront forwards to Amazon S3, CloudFront adds one or both of the following:

- An `If-Match` or `If-None-Match` header that contains the `ETag` value for the expired version of the object.
- An `If-Modified-Since` header that contains the `LastModified` value for the expired version of the object.

Amazon S3 uses this information to determine whether the object has been updated and, therefore, whether to return the entire object to CloudFront or to return only an HTTP 304 status code (not modified).

Cookies

Amazon S3 doesn't process cookies. If you configure a cache behavior to forward cookies to an Amazon S3 origin, CloudFront forwards the cookies, but Amazon S3 ignores them.

Cross-Origin Resource Sharing (CORS)

If you want CloudFront to respect Amazon S3 cross-origin resource sharing settings, configure CloudFront to forward the `Origin` header to Amazon S3. For more information, see Configuring CloudFront to Cache Objects Based on Request Headers (p. 78).

HTTP Methods

If you configure CloudFront to process all of the HTTP methods that it supports, CloudFront accepts the following requests from viewers and forwards them to your Amazon S3 origin:

- `DELETE`
- `GET`
- `HEAD`
- `OPTIONS`
- `PATCH`
- `POST`
- `PUT`

CloudFront caches responses to `GET` and `HEAD` requests, and does not cache responses to requests that use the other methods.

If you want to use multi-part uploads to add objects to an Amazon S3 bucket, you must add a CloudFront origin access identity to your distribution and grant the origin access identity the applicable permissions. For more information, see Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content (p. 123).

> **Caution**
> If you configure CloudFront to accept and forward to Amazon S3 all of the HTTP methods that CloudFront supports, you must create a CloudFront origin access identity to restrict access to your Amazon S3 content and grant the origin access identity the applicable permissions. For example, if you configure CloudFront to accept and forward these methods because you want to use `PUT`, you must configure Amazon S3 bucket policies or ACLs to handle `DELETE` requests appropriately so viewers can't delete resources that you don't want them to. For more information, see Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content (p. 123).

For information about the operations supported by Amazon S3, see the Amazon S3 documentation.

HTTP Request Headers that CloudFront Removes or Updates

CloudFront removes or updates the following header fields before forwarding requests to your Amazon S3 origin:

- `Accept`
- `Accept-Charset`
- `Accept-Encoding`: If the value contains `gzip`, CloudFront forwards `Accept-Encoding: gzip` to your Amazon S3 origin. If the value does not contain `gzip`, CloudFront removes the `Accept-Encoding` header field before forwarding the request to your origin.
- `Accept-Language`
- `Authorization`:
  - `GET` and `HEAD` requests: CloudFront removes the `Authorization` header field before forwarding the request to your origin.
  - `DELETE`, `OPTIONS`, `PATCH`, `POST`, and `PUT` requests: CloudFront does not remove the header field before forwarding the request to your origin.

- `Connection`: CloudFront replaces this header with `Connection: Keep-Alive` before forwarding the request to your Amazon S3 origin.
- `Cookie`: If you configure CloudFront to forward cookies, it will forward the `Cookie` header field to your Amazon S3 origin. If you don't, CloudFront removes the `Cookie` header field. For more information, see Configuring CloudFront to Cache Objects Based on Cookies (p. 76).
- `Expect`
- `Host`: CloudFront sets the value to the name of the Amazon S3 bucket that is associated with the requested object.
- `Proxy-Authorization`
- `Referer`
- `TE`
- `Upgrade`
- `User-Agent`: CloudFront replaces the value of this header field with `Amazon CloudFront`.

Maximum Length of a Request and Maximum Length of a URL

The maximum length of a request, including the path, the query string (if any), and headers, is 20480 bytes.

CloudFront constructs a URL from the request. The maximum length of this URL is 8192 bytes.

If a request or a URL exceeds these limits, CloudFront drops the request.

Protocols

CloudFront forwards HTTP or HTTPS requests to the origin server based on the protocol of the viewer request, either HTTP or HTTPS.

Query Strings

For web distributions, you can configure whether CloudFront forwards query string parameters to your Amazon S3 origin. For RTMP distributions, CloudFront does not forward query string parameters. For more information, see Configuring CloudFront to Cache Based on Query String Parameters (p. 75).

Request Timeout

When CloudFront requests data from your Amazon S3 origin, if Amazon S3 doesn't respond within 30 seconds or stops responding for 30 seconds, CloudFront drops the connection and makes two additional attempts to contact the origin. If the origin doesn't reply during the third attempt, CloudFront doesn't try again until it receives another request for content on the same origin.

How CloudFront Processes Responses from Your Amazon S3 Origin Server

Topics

- Canceled Requests (p. 105)
- HTTP Response Headers that CloudFront Removes or Updates (p. 105)
- Maximum File Size (p. 105)
- Redirects (p. 105)

Canceled Requests

If an object is not in the edge cache, and if a viewer terminates a session (for example, closes a browser) after CloudFront gets the object from your origin but before it can deliver the requested object, CloudFront does not cache the object in the edge location.

HTTP Response Headers that CloudFront Removes or Updates

CloudFront removes or updates the following header fields before forwarding the response from your Amazon S3 origin to the viewer:

- `Set-Cookie`: If you configure CloudFront to forward cookies, it will forward the `Set-Cookie` header field to clients. For more information, see Configuring CloudFront to Cache Objects Based on Cookies (p. 76).
- `Trailer`
- `Transfer-Encoding`: If your Amazon S3 origin returns this header field, CloudFront sets the value to `chunked` before returning the response to the viewer.
- `Upgrade`
- `Vary`
- `Via`: CloudFront sets the value to:

  ```
  Via: 1.1 alphanumeric-string.cloudfront.net (CloudFront)
  ```

  before returning the response to the viewer. For example:

  ```
  Via: 1.1 1026589cc7887e7a0dc7827b4example.cloudfront.net (CloudFront)
  ```

Maximum File Size

The maximum size of a response body that CloudFront will return to the viewer is 20 GB. This includes chunked transfer responses that don't specify the `Content-Length` header value.

Redirects

You can configure an Amazon S3 bucket to redirect all requests to another host name; this can be another Amazon S3 bucket or an HTTP server. If you configure a bucket to redirect all requests and if the bucket is the origin for a CloudFront distribution, we recommend that you configure the bucket to redirect all requests to a CloudFront distribution using either the domain name for the distribution (for example, d111111abcdef8.cloudfront.net) or an alternate domain name (a CNAME) that is associated with a distribution (for example, example.com). Otherwise, viewer requests bypass CloudFront, and the objects are served directly from the new origin.

> **Note**
> If you redirect requests to an alternate domain name, you must also update the DNS service for your domain by adding a CNAME record. For more information, see Using Alternate Domain Names (CNAMEs) (p. 29).

Here's what happens when you configure a bucket to redirect all requests:

1. A viewer (for example, a browser) requests an object from CloudFront.
2. CloudFront forwards the request to the Amazon S3 bucket that is the origin for your distribution.
3. Amazon S3 returns an HTTP status code 301 (Moved Permanently) as well as the new location.

4. CloudFront caches the redirect status code and the new location, and returns the values to the viewer. CloudFront does not follow the redirect to get the object from the new location.
5. The viewer sends another request for the object, but this time the viewer specifies the new location that it got from CloudFront:

   - If the Amazon S3 bucket is redirecting all requests to a CloudFront distribution, using either the domain name for the distribution or an alternate domain name, CloudFront requests the object from the Amazon S3 bucket or the HTTP server in the new location. When the new location returns the object, CloudFront returns it to the viewer and caches it in an edge location.
   - If the Amazon S3 bucket is redirecting requests to another location, the second request bypasses CloudFront. The Amazon S3 bucket or the HTTP server in the new location returns the object directly to the viewer, so the object is never cached in a CloudFront edge cache.

Request and Response Behavior for Custom Origins

Topics

- How CloudFront Processes and Forwards Requests to Your Custom Origin Server (p. 106)
- How CloudFront Processes Responses from Your Custom Origin Server (p. 113)

How CloudFront Processes and Forwards Requests to Your Custom Origin Server

For information about how CloudFront processes viewer requests and forwards the requests to your custom origin, see the applicable topic:

Topics

- Authentication (p. 107)
- Caching Duration and Minimum TTL (p. 107)
- Client IP Addresses (p. 107)
- Compression (p. 107)
- Conditional Requests (p. 107)
- Cookies (p. 108)
- Cross-Origin Resource Sharing (CORS) (p. 108)
- Encryption (p. 108)
- HTTP Methods (p. 108)
- HTTP Request Headers and CloudFront Behavior (p. 109)
- HTTP Version (p. 111)
- Maximum Length of a Request and Maximum Length of a URL (p. 111)
- Protocols (p. 111)
- Query Strings (p. 112)
- Request Timeout (p. 112)
- User-Agent Header (p. 112)

Authentication

For GET and HEAD requests, do not configure your origin server to request client authentication. CloudFront removes the Authorization header before forwarding requests to your origin.

For DELETE, OPTIONS, PATCH, POST, and PUT requests, if you configure CloudFront to forward the Authorization header to your origin, you can configure your origin server to request client authentication.

You can configure CloudFront to forward requests to your origin using either HTTP or HTTPS; for more information, see How to Require HTTPS for Communication between Viewers, CloudFront, and Your Origin (p. 169).

Caching Duration and Minimum TTL

For web distributions, to control how long your objects stay in a CloudFront cache before CloudFront forwards another request to your origin, you can:

- Configure your origin to add a Cache-Control or an Expires header field to each object.
- Specify a value for Minimum TTL in CloudFront cache behaviors.
- Use the default value of 24 hours.

For more information, see Specifying How Long Objects Stay in a CloudFront Edge Cache (Expiration) (p. 83).

Client IP Addresses

CloudFront forwards the client IP address to your origin in the X-Forwarded-For request header. The X-Forwarded-For request header looks like this:

```
X-Forwarded-For: 192.0.2.235
```

If the X-Forwarded-For request header contains two or more IP addresses, the first one is always the client IP address. The other addresses are for each successive proxy that passes along the request; the last IP address is for the proxy that forwarded the request to CloudFront:

```
X-Forwarded-For: client-IP-address, proxy-IP-address, another-proxy-IP-address
```

Compression

CloudFront forwards requests that have the Accept-Encoding field values "identity" and "gzip". For more information, see Serving Compressed Files (p. 98).

Conditional Requests

When CloudFront receives a request for an object that has expired from an edge cache, it forwards the request to the origin either to get the latest version of the object or to get confirmation from the origin that the CloudFront edge cache already has the latest version. Typically, when the origin last sent the object to CloudFront, it included an ETag value, a LastModified value, or both values in the response. In the new request that CloudFront forwards to the origin, CloudFront adds one or both of the following:

- An If-Match or If-None-Match header that contains the ETag value for the expired version of the object.
- An If-Modified-Since header that contains the LastModified value for the expired version of the object.

The origin uses this information to determine whether the object has been updated and, therefore, whether to return the entire object to CloudFront or to return only an HTTP 304 status code (not modified).

Cookies

You can configure CloudFront to forward cookies to your origin. For more information, see Configuring CloudFront to Cache Objects Based on Cookies (p. 76).

Cross-Origin Resource Sharing (CORS)

If you want CloudFront to respect cross-origin resource sharing settings, configure CloudFront to forward the `Origin` header to your origin. For more information, see Configuring CloudFront to Cache Objects Based on Request Headers (p. 78).

Encryption

CloudFront forwards HTTPS requests to the origin server using the SSLv3 or TLSv1 protocols and the following ciphers:

- ECDHE-RSA-AES128-GCM-SHA256
- ECDHE-RSA-AES128-SHA256
- ECDHE-RSA-AES128-SHA
- ECDHE-RSA-AES256-GCM-SHA384
- ECDHE-RSA-AES256-SHA384
- ECDHE-RSA-AES256-SHA
- AES128-GCM-SHA256
- AES256-GCM-SHA384
- AES128-SHA256
- AES256-SHA
- AES128-SHA
- RC4-MD5

If your origin server does not support at least one of these ciphers, CloudFront cannot establish an SSL connection to your origin.

When establishing an HTTPS connection to the origin, CloudFront adds a Server Name Indication (SNI) extension and includes the value of the applicable **Origin Domain Name** for your distribution. For more information about SNI, see Section 3.1 of RFC 4366, Transport Layer Security (TLS) Extensions.

HTTP Methods

If you configure CloudFront to process all of the HTTP methods that it supports, CloudFront accepts the following requests from viewers and forwards them to your custom origin:

- `DELETE`
- `GET`
- `HEAD`
- `OPTIONS`
- `PATCH`
- `POST`
- `PUT`

CloudFront caches responses to GET and HEAD requests, and does not cache responses to requests that use the other methods.

For information about configuring whether your custom origin processes these methods, see the documentation for your origin.

> **Caution**
> If you configure CloudFront to accept and forward to your origin all of the HTTP methods that CloudFront supports, configure your origin server to handle all methods. For example, if you configure CloudFront to accept and forward these methods because you want to use POST, you must configure your origin server to handle DELETE requests appropriately so viewers can't delete resources that you don't want them to. For more information, see the documentation for your HTTP server.

HTTP Request Headers and CloudFront Behavior

The following table lists HTTP request headers and, for each header, explains the following:

- CloudFront behavior if you don't configure CloudFront to forward the header to your origin, which causes CloudFront to cache your objects based on header values.
- Whether you can configure CloudFront to cache objects based on header values for that header.

  You can configure CloudFront to cache objects based on values in the Date and User-Agent headers, but we don't recommend it. These headers have a lot of possible values, and caching based on their values would cause CloudFront to forward significantly more requests to your origin.

For more information about caching based on header values, see Configuring CloudFront to Cache Objects Based on Request Headers (p. 78).

Header	Behavior If You Don't Configure CloudFront to Cache Based on Header Values	Caching Based on Header Values Is Supported
Customer-defined headers	CloudFront forwards the headers to your origin.	Yes
Accept	CloudFront removes the header.	Yes
Accept-Charset	CloudFront removes the header.	Yes
Accept-Encoding	If the value contains gzip, CloudFront forwards Accept-Encoding: gzip to your origin. If the value does not contain gzip, CloudFront removes the Accept-Encoding header field before forwarding the request to your origin.	No
Accept-Language	CloudFront removes the header.	Yes
Authorization	GET and HEAD requests: CloudFront removes the Authorization header field before forwarding the request to your origin. DELETE, OPTIONS, PATCH, POST, and PUT requests: CloudFront does not remove the header field before forwarding the request to your origin.	Yes
Cache-Control	CloudFront forwards the header to your origin.	Yes

Header	Behavior If You Don't Configure CloudFront to Cache Based on Header Values	Caching Based on Header Values Is Supported
CloudFront-Forwarded-Proto	CloudFront does not add the header before forwarding the request to your origin. For more information, see Configuring CloudFront to Cache Objects Based on the Protocol of the Request (p. 81).	Yes
CloudFront-Is-Desktop-Viewer	CloudFront does not add the header before forwarding the request to your origin. For more information, see Configuring CloudFront to Cache Objects Based on the Device Type (p. 80).	Yes
CloudFront-Is-Mobile-Viewer	CloudFront does not add the header before forwarding the request to your origin. For more information, see Configuring CloudFront to Cache Objects Based on the Device Type (p. 80).	Yes
CloudFront-Is-Tablet-Viewer	CloudFront does not add the header before forwarding the request to your origin. For more information, see Configuring CloudFront to Cache Objects Based on the Device Type (p. 80).	Yes
CloudFront-Viewer-Country	CloudFront does not add the header before forwarding the request to your origin.	Yes
Connection	CloudFront replaces this header with `Connection: Keep-Alive` before forwarding the request to your origin.	No
Content-Length	CloudFront forwards the header to your origin.	Yes
Content-MD5	CloudFront forwards the header to your origin.	Yes
Content-Type	CloudFront forwards the header to your origin.	Yes
Cookie	If you configure CloudFront to forward cookies, it will forward the `Cookie` header field to your origin. If you don't, CloudFront removes the `Cookie` header field. For more information, see Configuring CloudFront to Cache Objects Based on Cookies (p. 76).	No
Date	CloudFront forwards the header to your origin.	Yes, but not recommended
Expect	CloudFront removes the header.	Yes
From	CloudFront forwards the header to your origin.	Yes
Host	CloudFront sets the value to the domain name of the origin that is associated with the requested object.	Yes
If-Match	CloudFront forwards the header to your origin.	Yes

Header	Behavior If You Don't Configure CloudFront to Cache Based on Header Values	Caching Based on Header Values Is Supported
If-Modified-Since	CloudFront forwards the header to your origin.	Yes
If-None-Match	CloudFront forwards the header to your origin.	Yes
If-Range	CloudFront forwards the header to your origin.	Yes
If-Unmodified-Since	CloudFront forwards the header to your origin.	Yes
Max-Forward	CloudFront forwards the header to your origin.	Yes
Origin	CloudFront forwards the header to your origin.	Yes
Pragma	CloudFront forwards the header to your origin.	Yes
Proxy-Authorization	CloudFront removes the header.	No
Range	CloudFront forwards the header to your origin. For more information, see How CloudFront Processes Partial Requests for an Object (Range GETs) (p. 96).	Yes
Referer	CloudFront removes the header.	Yes
TE	CloudFront removes the header.	No
Upgrade	CloudFront removes the header.	No
User-Agent	CloudFront replaces the value of this header field with `Amazon CloudFront`. If you want CloudFront to cache your content based on the device the user is using, see Configuring CloudFront to Cache Objects Based on the Device Type (p. 80).	Yes, but not recommended
Via	CloudFront forwards the header to your origin.	Yes
Warning	CloudFront forwards the header to your origin.	Yes

HTTP Version

CloudFront forwards requests to your custom origin using HTTP/1.1.

Maximum Length of a Request and Maximum Length of a URL

The maximum length of a request, including the path, the query string (if any), and headers, is 20480 bytes.

CloudFront constructs a URL from the request. The maximum length of this URL is 8192 bytes.

If a request or a URL exceeds these limits, CloudFront drops the request.

Protocols

CloudFront forwards HTTP or HTTPS requests to the origin server based on the following:

- The protocol of the request that the viewer sends to CloudFront, either HTTP or HTTPS.
- The value of the **Origin Protocol Policy** field in the CloudFront console or, if you're using the CloudFront API, the `OriginProtocolPolicy` element in the `DistributionConfig` complex type. In the CloudFront console, the options are **HTTP Only** and **Match Viewer**.

If you specify **HTTP Only**, CloudFront forwards requests to the origin server using only the HTTP protocol, regardless of the protocol in the viewer request.

If you specify **Match Viewer**, CloudFront forwards requests to the origin server using the protocol in the viewer request. Note that CloudFront caches the object only once even if viewers make requests using both HTTP and HTTPS protocols.

> **Caution**
> If the viewer request uses the HTTPS protocol, and if the origin server returns an invalid certificate or a self-signed certificate, CloudFront drops the TCP connection.

If you aren't sure which protocol to use, we recommend that you specify HTTP only.

For information about how to update a distribution using the CloudFront console, see Listing, Viewing, and Updating CloudFront Distributions (p. 27). For information about how to update a distribution using the CloudFront API, go to PUT Distribution Config in the *Amazon CloudFront API Reference*.

Query Strings

You can configure whether CloudFront forwards query string parameters to your origin. For more information, see Configuring CloudFront to Cache Based on Query String Parameters (p. 75).

Request Timeout

When CloudFront requests data from your origin, if the origin doesn't respond within 30 seconds or stops responding for 30 seconds, CloudFront drops the connection and makes two additional attempts to contact the origin. If the origin doesn't reply during the third attempt, CloudFront doesn't try again until it receives another request for content on the same origin.

User-Agent Header

If you want CloudFront to cache different versions of your objects based on the device a user is using to view your content, we recommend that you configure CloudFront to forward the applicable headers to your custom origin:

- `CloudFront-Is-Mobile-Viewer`
- `CloudFront-Is-Tablet-Viewer`
- `CloudFront-Is-Desktop-Viewer`

Based on the value of the `User-Agent` header, CloudFront sets the value of these headers to `true` or `false` before forwarding the request to your origin. If a device falls into more than one category, more than one value might be `true`. For example, for some tablet devices, CloudFront might set both `CloudFront-Is-Mobile-Viewer` and `CloudFront-Is-Tablet-Viewer` to `true`. For more information about configuring CloudFront to cache based on request headers, see Configuring CloudFront to Cache Objects Based on Request Headers (p. 78).

You can configure CloudFront to cache objects based on values in the `User-Agent` header, but we don't recommend it. The `User-Agent` header has a lot of possible values, and caching based on those values would cause CloudFront to forward significantly more requests to your origin.

If you do not configure CloudFront to cache objects based on values in the `User-Agent` header, CloudFront CloudFront adds a `User-Agent` header with the following value before it forwards a request to your origin:

```
User-Agent = Amazon CloudFront
```

CloudFront adds this header regardless of whether the request from the viewer included a `User-Agent` header. If the request from the viewer includes a `User-Agent` header, CloudFront removes it.

How CloudFront Processes Responses from Your Custom Origin Server

For information about how CloudFront processes responses from custom origin servers, see the applicable topic:

Topics

- Caching (p. 113)
- Canceled Requests (p. 113)
- Content Negotiation (p. 113)
- Cookies (p. 113)
- Dropped TCP Connections (p. 114)
- HTTP Response Headers that CloudFront Removes or Updates (p. 114)
- Maximum File Size (p. 114)
- Origin Unavailable (p. 114)
- Redirects (p. 115)
- Transfer Encoding (p. 115)

Caching

- Ensure that the origin server sets valid and accurate values for the `Date` and `Last-Modified` header fields.
- If requests from viewers include the `If-Match` or `If-None-Match` request header fields, set the `ETag` response header field. If you do not specify an `ETag` value, CloudFront ignores subsequent `If-Match` or `If-None-Match` headers.

Canceled Requests

If an object is not in the edge cache, and if a viewer terminates a session (for example, closes a browser) after CloudFront gets the object from your origin but before it can deliver the requested object, CloudFront does not cache the object in the edge location.

Content Negotiation

The only acceptable value for the `Vary` header is `Accept-Encoding`. CloudFront ignores other values.

Cookies

If you enable cookies for a cache behavior, and if the origin returns cookies with an object, CloudFront caches both the object and the cookies. Note that this reduces cacheability for an object. For more information, see Configuring CloudFront to Cache Objects Based on Cookies (p. 76).

Dropped TCP Connections

If the TCP connection between CloudFront and your origin drops while your origin is returning an object to CloudFront, CloudFront behavior depends on whether your origin included a `Content-Length` header in the response:

- **Content-Length header:** CloudFront returns the object to the viewer as it gets the object from your origin. However, if the value of the `Content-Length` header doesn't match the size of the object, CloudFront doesn't cache the object.
- **Transfer-Encoding: Chunked:** CloudFront returns the object to the viewer as it gets the object from your origin. However, if the chunked response is not complete, CloudFront does not cache the object.
- **No Content-Length header:** CloudFront returns the object to the viewer and caches it, but the object may not be complete. Without a `Content-Length` header, CloudFront cannot determine whether the TCP connection was dropped accidentally or on purpose.

We recommend that you configure your HTTP server to add a `Content-Length` header to prevent CloudFront from caching partial objects.

HTTP Response Headers that CloudFront Removes or Updates

CloudFront removes or updates the following header fields before forwarding the response from your origin to the viewer:

- `Set-Cookie`: If you configure CloudFront to forward cookies, it will forward the `Set-Cookie` header field to clients. For more information, see Configuring CloudFront to Cache Objects Based on Cookies (p. 76).
- `Trailer`
- `Transfer-Encoding`: If your origin returns this header field, CloudFront sets the value to `chunked` before returning the response to the viewer.
- `Upgrade`
- `Vary`
- `Via`: Regardless of whether your origin returns this header field to CloudFront, CloudFront sets the value to:

  ```
  Via: 1.1 alphanumeric-string.cloudfront.net (CloudFront)
  ```

  before returning the response to the viewer. For example:

  ```
  Via: 1.1 1026589cc7887e7a0dc7827b4example.cloudfront.net (CloudFront)
  ```

Maximum File Size

The maximum size of a response body that CloudFront will return to the viewer is 20 GB. This includes chunked transfer responses that don't specify the `Content-Length` header value.

Origin Unavailable

If your origin server is unavailable and CloudFront gets a request for an object that is in the edge cache but that has expired (for example, because the period of time specified in the `Cache-Control max-age` directive has passed), CloudFront either serves the expired version of the object or serves a custom error page. For more information, see How CloudFront Processes and Caches HTTP 4xx and 5xx Status Codes (p. 115).

In some cases, an object that is seldom requested is evicted and is no longer available in the edge cache. CloudFront can't serve an object that has been evicted.

Redirects

If you change the location of an object on the origin server, you can configure your web server to redirect requests to the new location. After you configure the redirect, the first time a viewer submits a request for the object, CloudFront Front sends the request to the origin, and the origin responds with a redirect (for example, `302 Moved Temporarily`). CloudFront caches the redirect and returns it to the viewer. CloudFront does not follow the redirect.

You can configure your web server to redirect requests to one of the following locations:

- The new URL of the object on the origin server. When the viewer follows the redirect to the new URL, the viewer bypasses CloudFront and goes straight to the origin. As a result, we recommend that you not redirect requests to the new URL of the object on the origin.
- The new CloudFront URL for the object. When the viewer submits the request that contains the new CloudFront URL, CloudFront gets the object from the new location on your origin, caches it at the edge location, and returns the object to the viewer. Subsequent requests for the object will be served by the edge location. This avoids the latency and load associated with viewers requesting the object from the origin. However, every new request for the object will incur charges for two requests to CloudFront.

Transfer Encoding

CloudFront supports only the `chunked` value of the `Transfer-Encoding` header. If your origin returns `Transfer-Encoding: chunked`, CloudFront CloudFront returns the object to the client as the object is received at the edge location, and caches the object in chunked format for subsequent requests.

We recommend that you use chunked encoding if the content length of your response cannot be predetermined. For more information, see Dropped TCP Connections (p. 114).

How CloudFront Processes and Caches HTTP 4xx and 5xx Status Codes

Topics

- How CloudFront Processes Errors When You Have Configured Custom Error Pages (p. 116)
- How CloudFront Processes Errors When You Have Not Configured Custom Error Pages (p. 116)
- HTTP 4xx and 5xx Status Codes that CloudFront Caches (p. 117)

When CloudFront requests an object from your Amazon S3 bucket or custom origin server, your origin sometimes returns an HTTP 4xx or 5xx status code, which indicates an error has occurred. CloudFront behavior depends on:

- Whether you have configured custom error pages.
- Whether you have configured how long you want CloudFront to cache error responses from your origin (error caching minimum TTL).
- The status code.
- For 5xx status codes, whether the requested object is currently in the CloudFront edge cache.

For information about settings for custom error pages in the CloudFront console, see Custom Error Pages and Error Caching (p. 53). For information about the error caching minimum TTL in the CloudFront console, see Error Caching Minimum TTL (p. 54).

For a list of the HTTP status codes that CloudFront caches, see HTTP 4xx and 5xx Status Codes that CloudFront Caches (p. 117).

If you have enabled logging, CloudFront writes the results to the logs regardless of the HTTP status code.

How CloudFront Processes Errors When You Have Configured Custom Error Pages

When your origin returns an HTTP 4xx or 5xx status code, the requested object is *not* in the edge cache, and you have configured custom error pages, CloudFront does the following:

1. In the CloudFront edge cache that received the original request, CloudFront checks your distribution configuration and gets the path of the custom error page that corresponds with the status code that your origin server returned.
2. CloudFront finds the first cache behavior in your distribution that has a path pattern that matches the path of the custom error page.
3. The CloudFront edge location sends a request for the custom error page to the origin that is specified in the cache behavior.
4. The origin returns the custom error page to the edge location.
5. CloudFront returns the custom error page to the viewer that made the request, and also caches the custom error page for the amount of time specified by the error caching minimum TTL (five minutes by default).
6. After the error caching minimum TTL has elapsed, CloudFront tries again to get the requested object by forwarding another request to your origin.

When your origin returns an HTTP 5xx status code, the requested object *is* in the edge cache, and you have configured custom error pages, CloudFront does the following:

1. CloudFront serves the object even though it has expired. For the duration of the error caching minimum TTL, CloudFront continues to respond to viewer requests by serving the object from the edge cache.
2. After the error caching minimum TTL has elapsed, CloudFront tries again to get the requested object by forwarding another request to your origin. Note that if the object is not requested frequently, CloudFront may evict it from the edge cache while your origin server is still returning 5xx responses. For information about how long objects stay in CloudFront edge caches, see Specifying How Long Objects Stay in a CloudFront Edge Cache (Expiration) (p. 83).

How CloudFront Processes Errors When You Have Not Configured Custom Error Pages

When your origin either returns an HTTP 4xx status code or returns an HTTP 5xx status code when the requested object is *not* in the edge cache, and you have not configured custom error pages, CloudFront does the following:

1. CloudFront returns the 4xx or 5xx status code to the viewer.
2. CloudFront also caches the status code in the edge cache that received the request.
3. For the duration of the error caching minimum TTL (five minutes by default), CloudFront responds to subsequent viewer requests for the same object with the cached 4xx or 5xx status code.

4. After the error caching minimum TTL has elapsed, CloudFront tries again to get the requested object by forwarding another request to your origin.

When your origin returns an HTTP 5xx status code, the requested object *is* in the edge cache, and you have not configured custom error pages, CloudFront does the following:

1. CloudFront serves the object even though it has expired. For the duration of the error caching minimum TTL, CloudFront continues to respond to viewer requests by serving the object from the edge cache.
2. After the error caching minimum TTL has elapsed, CloudFront tries again to get the requested object by forwarding another request to your origin. Note that if the object is not requested frequently, CloudFront may evict it from the edge cache while your origin server is still returning 5xx responses. For information about how long objects stay in CloudFront edge caches, see Specifying How Long Objects Stay in a CloudFront Edge Cache (Expiration) (p. 83).

HTTP 4xx and 5xx Status Codes that CloudFront Caches

CloudFront caches the following HTTP 4xx and 5xx status codes returned by Amazon S3 or your custom origin server. If you have configured a custom error page for an HTTP status code, CloudFront caches the custom error page.

400	Bad Request
403	Forbidden
404	Not Found
405	Method Not Allowed
414	Request-URI Too Large
500	Internal Service Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Time-out

Serving Private Content through CloudFront

Many companies that distribute content via the Internet want to restrict access to documents, business data, media streams, or content that is intended for selected users, for example, users who have paid a fee. To securely serve this private content using CloudFront, you can:

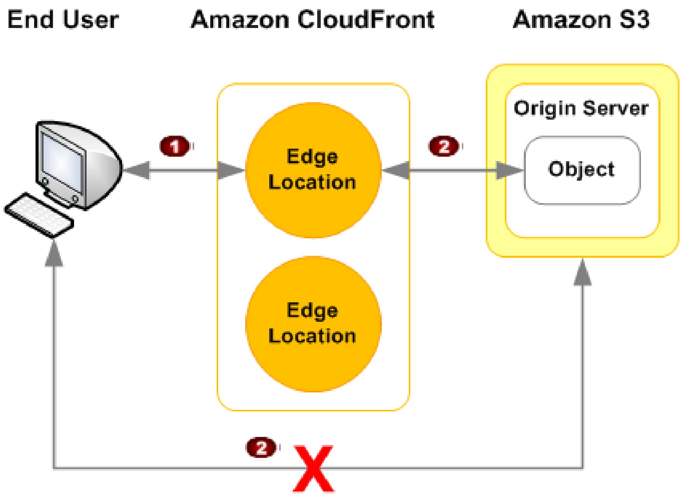
- Require that your users use special CloudFront signed URLs to access your content, not the standard CloudFront public URLs.
- Require that your users access your Amazon S3 content using CloudFront URLs, not Amazon S3 URLs.

Overview of Private Content

You can control end-user access to your private content in two ways:

1	**You can restrict access to objects in CloudFront edge caches:** You can configure CloudFront to require that end users access your objects using special **signed URLs**. You then create the signed URLs (either manually or programmatically) and distribute them to your users. When you create signed URLs for your objects, you can specify: • An ending date and time, after which the URL is no longer valid. • (Optional) The date and time that the URL becomes valid. • (Optional) The IP address or range of addresses of the computers that can be used to access your content. One part of a signed URL is hashed and signed using the private key from a public/private key pair. When someone uses a signed URL to access an object, CloudFront compares the signed and unsigned portions of the URL. If they don't match, CloudFront doesn't serve the object.

2	**You can restrict access to objects in your Amazon S3 bucket:** You can secure the content in your Amazon S3 bucket so end users can access it using CloudFront URLs but cannot access it using Amazon S3 URLs. This prevents anyone from bypassing CloudFront and using the Amazon S3 URL to access the content to which you're trying to restrict access. To require that users use CloudFront URLs, you: • Create a special CloudFront user called an **origin access identity**. • Give the origin access identity permission to read the objects in your bucket. • Remove permission for anyone else to read the objects.



How Private Content Works

Here's an overview of how you use private content to secure access to your Amazon S3 content. Later, we go into greater detail for each step.

> **Note**
> To set up private content, you must use the CloudFront console or CloudFront API version 2009-09-09 or later.

1. You secure your content in Amazon S3 to prevent anyone from bypassing CloudFront and using the Amazon S3 URL to access the content to which you're trying to restrict access. This step is optional, but it's a good idea in case someone learns the Amazon S3 URLs for your content.

   a. Create an origin access identity, which is a special CloudFront user.
   b. Associate the origin access identity with your distribution. (For web distributions, you associate the origin access identity with origins, so you can secure all or just some of your Amazon S3 content.)
   c. Change permissions in Amazon S3 so only the origin access identity can access your objects.

   For more information, see Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content (p. 123).
2. In your CloudFront distribution, specify one or more trusted signers, which are the AWS accounts that you want to have permission to create signed URLs.

For more information, see Specifying the AWS Accounts That Can Create Signed URLs (Trusted Signers) (p. 128).

3. You develop your application to create signed URLs for the objects or parts of your application for which signed URLs are required.

   For more information about signed URLs, see Overview of Signed URLs (p. 134).
4. An end user requests an object for which you want to require signed URLs.
5. Your application verifies that the end user is entitled to access the object: they've signed in, they've paid for access to the content, or they've met some other requirement for access.
6. Your application creates and returns a signed URL to the user.
7. The signed URL allows the user to download or stream the content.

   This step is automatic; the user usually doesn't have to do anything additional to access the content. For example, if a user is accessing your content in a web browser, your application returns the signed URL to the browser. The browser immediately uses the signed URL to access the object in the CloudFront edge cache without any intervention from the user.
8. CloudFront confirms that the URL hasn't been tampered with and that it's still valid. For example, if you specified a beginning and ending date and time for the URL, CloudFront confirms that the user is trying to access your content during the time period that you want to allow access. If the URL is valid, CloudFront performs the standard operations: determines whether the object is already in the edge cache, forwards the request to the origin if necessary, and returns the object to the user.

Using an HTTP Server for Private Content

You can use signed URLs for any CloudFront distribution, regardless of whether the origin is an Amazon S3 bucket or an HTTP server. However, for CloudFront to access your objects on an HTTP server, the objects must remain publicly accessible. Because the objects are publicly accessible, anyone who has the URL for an object on your HTTP server can access the object without the protection provided by CloudFront signed URLs. If you use signed URLs and your origin is an HTTP server, do not give the URLs for the objects on your HTTP server to your customers or to others outside your organization.

Choosing How Long Signed URLs Are Valid

You can distribute private content using a signed URL that is valid for only a short time—possibly for as little as a few minutes. Signed URLs that are valid for such a short period are good for distributing content on-the-fly to an end user for a limited purpose, such as distributing movie rentals or music downloads to customers on demand. If your signed URLs will be valid for just a short period, you'll probably want to generate them automatically using an application that you develop. When the user starts to download an object or starts to play a media file, CloudFront compares the expiration time in the URL with the current time to determine whether the URL is still valid.

You can also distribute private content using a signed URL that is valid for a longer time, possibly for years. Signed URLs that are valid for a longer period are useful for distributing private content to known end users, such as distributing a business plan to investors or distributing training materials to employees. You can develop an application to generate these longer-term signed URLs for you, or you can use one of the third-party GUI tools listed in Tools for Configuring Private Content (p. 285).

Sample Code and Third-Party Tools

For sample code that creates the hashed and signed part of signed URLs, see the following topics:

- Create a URL Signature Using Perl (p. 153)
- Create a URL Signature Using PHP (p. 154)
- Create a URL Signature Using C# and the .NET Framework (p. 157)

- Create a URL Signature Using Java (p. 165)

Additional sample code for creating signed URLs is available on the Amazon CloudFront Sample Code & Libraries page.

For information about third-party tools that support private content, including creating signed URLs, see Tools for Configuring Private Content (p. 285).

Task List: Serving Private Content

To configure CloudFront to serve private content, perform the following tasks.

1. *Optional:* Configure your CloudFront distribution and your Amazon S3 bucket to require that your users access your Amazon S3 content using only CloudFront URLs. For more information, see Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content (p. 123).
2. Specify the AWS accounts that you want to use to create signed URLs. For more information, see Specifying the AWS Accounts That Can Create Signed URLs (Trusted Signers) (p. 128).
3. Create signed URLs, either manually or programmatically, to give to authorized end users. For more information, see Overview of Signed URLs (p. 134).

Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content

Topics

- Creating a CloudFront Origin Access Identity and Adding it to Your Distribution (p. 123)
- Granting the Origin Access Identity Permission to Read Objects in Your Amazon S3 Bucket (p. 125)

Typically, if you're using an Amazon S3 bucket as the origin for a CloudFront distribution, you grant everyone permission to read the objects in your bucket. This allows anyone to access your objects using either the CloudFront URL or the Amazon S3 URL. CloudFront doesn't expose Amazon S3 URLs, but your users may have those URLs if your application serves any objects directly from Amazon S3 or if anyone gives out direct links to specific objects in Amazon S3.

If you want to use CloudFront signed URLs to provide access to objects in your Amazon S3 bucket, you probably also want to prevent users from accessing your Amazon S3 objects using Amazon S3 URLs. If users access your objects directly in Amazon S3, they bypass the controls provided by CloudFront signed URLs, including control over when a URL expires and control over which IP addresses can be used to access the objects. In addition, if users access objects using both CloudFront URLs and Amazon S3 URLs, CloudFront access logs are less useful because they're incomplete.

You restrict access to Amazon S3 content by creating an origin access identity, which is a special CloudFront user. You change Amazon S3 permissions to give the origin access identity permission to access your objects, and to remove permissions from everyone else. When your users access your Amazon S3 objects using CloudFront URLs, the CloudFront origin access identity gets the objects on your users' behalf. If your users try to access objects using Amazon S3 URLs, they're denied access. The origin access identity has permission to access objects in your Amazon S3 bucket, but users don't.

> **Note**
> To create origin access identities, you must use the CloudFront console or CloudFront API version 2009-09-09 or later.

To ensure that your users access your objects using only CloudFront URLs, regardless of whether the URLs are signed, perform the following tasks:

1. Create an origin access identity and add it to your distribution. For more information, see Creating a CloudFront Origin Access Identity and Adding it to Your Distribution (p. 123).

   > **Note**
   > You can also create an origin access identity and add it to your distribution when you create the distribution.

2. Change the permissions either on your Amazon S3 bucket or on the objects in your bucket so only the origin access identity has read permission (or read and download permission).

   For more information, see Granting the Origin Access Identity Permission to Read Objects in Your Amazon S3 Bucket (p. 125).

Creating a CloudFront Origin Access Identity and Adding it to Your Distribution

An AWS account can have up to 100 CloudFront origin access identities. However, you can add an origin access identity to as many distributions as you want, so one origin access identity is usually sufficient.

If you didn't create an origin access identity and add it to your distribution when you created the distribution, you can create and add one now using either the CloudFront console or the CloudFront API:

- **If you're using the CloudFront console:** You can create an origin access identity and add it to your distribution at the same time. For more information, see Creating an Origin Access Identity and Adding it to Your Distribution Using the CloudFront Console (p. 124).
- **If you're using the CloudFront API:** You create an origin access identity and then you add it to your distribution. Perform the procedure in each of the following topics:
  - Creating an Origin Access Identity Using the CloudFront API (p. 125)
  - Adding an Origin Access Identity to Your Distribution Using the CloudFront API (p. 125)

Creating an Origin Access Identity and Adding it to Your Distribution Using the CloudFront Console

If you didn't create an origin access identity when you created your distribution, perform the following procedure.

To create a CloudFront origin access identity using the CloudFront console

1. Sign in to the AWS Management Console and open the Amazon CloudFront console at https://console.aws.amazon.com/cloudfront/.
2. Click the i icon for the distribution to which you want to add an origin access identity.
3. Change to edit mode:
   - **Web distributions:** Click the **Origins** tab, click the origin that you want to edit, and click **Edit**. You can only create an origin access identity for origins for which **Origin Type** is **S3 Origin**.
   - **RTMP distributions:** Click **Edit**.
4. For **Restrict Bucket Access**, click **Yes**.
5. If you already have an origin access identity that you want to use, click **Use an Existing Identity**. Then select the identity in the **Your Identities** list.

   Note
   If you already have an origin access identity, we recommend that you reuse it to simplify maintenance.

   If you want to create an identity, click **Create a New Identity**. Then enter a description for the identity in the **Comment** field.
6. If you want CloudFront to automatically give the origin access identity permission to read the objects in the Amazon S3 bucket specified in **Origin Domain Name**, click **Yes, Update Bucket Policy**.

   Important
   If you click **Yes, Update Bucket Policy**, CloudFront updates bucket permissions to grant the specified origin access identity the permission to read objects in your bucket. However, CloudFront does not remove existing permissions. If users currently have permission to access the objects in your bucket using Amazon S3 URLs, they will still have that permission after CloudFront updates your bucket permissions. To view or remove existing bucket permissions, use a method provided by Amazon S3. For more information, see Granting the Origin Access Identity Permission to Read Objects in Your Amazon S3 Bucket (p. 125).

   If you want to manually update permissions on your Amazon S3 bucket, click **No, I Will Update Permissions**.
7. Click **Yes, Edit**.

8. If you're adding an origin access identity to a web distribution and you have more than one origin, repeat Step 3 through Step 7 as applicable.

Creating an Origin Access Identity Using the CloudFront API

If you already have an origin access identity and you want to reuse it instead of creating another one, skip to Adding an Origin Access Identity to Your Distribution Using the CloudFront API (p. 125).

To create a CloudFront origin access identity using the CloudFront API, use the `POST Origin Access Identity` API action. The response includes an `Id` and an `S3CanonicalUserId` for the new origin access identity. Make note of these values because you will use them later in the process:

- **Id element:** You use the value of the `Id` element to associate an origin access ID with your distribution.
- **S3CanonicalUserId element:** You use the value of the `S3CanonicalUserId` element when you give CloudFront access to your Amazon S3 bucket or objects.

For more information about the `POST Origin Access Identity API` action, go to POST Origin Access Identity in the *Amazon CloudFront API Reference*. For a list of other actions that you can perform on origin access identities, go to Actions on Origin Access Identities, also in the *Amazon CloudFront API Reference*.

Adding an Origin Access Identity to Your Distribution Using the CloudFront API

You can use the CloudFront API to add a CloudFront origin access identity to an existing distribution or to create a new distribution that includes an origin access identity. In either case, include an `OriginAccessIdentity` element. This element contains the value of the `Id` element that the `POST Origin Access Identity` API action returned when you created the origin access identity. For web distributions, add the `OriginAccessIdentity` element to one or more origins. For RTMP distributions, add the `OriginAccessIdentity` element to the distribution.

See the applicable topic in the *Amazon CloudFront API Reference*:

- **Create a new web distribution:** POST Distribution
- **Update an existing web distribution:** PUT Distribution Config
- **Create a new RTMP distribution:** POST Streaming Distribution
- **Update an existing RTMP distribution:** PUT Streaming Distribution Config

Granting the Origin Access Identity Permission to Read Objects in Your Amazon S3 Bucket

When you create or update a distribution, you can add an origin access identity and automatically update the bucket policy to give the origin access identity permission to access your bucket. Alternatively, you can choose to manually change the bucket policy or change ACLs, which control permissions on individual objects in your bucket.

Whichever method you use, you should still review the bucket policy for your bucket and review the permissions on your objects to ensure that:

- CloudFront can access objects in the bucket on behalf of users who are requesting your objects using CloudFront URLs.
- Users can't use Amazon S3 URLs to access your objects.

Caution

If you configure CloudFront to accept and forward to Amazon S3 all of the HTTP methods that CloudFront supports, create a CloudFront origin access identity to restrict access to your Amazon S3 content, and grant the origin access identity the applicable permissions. For example, if you configure CloudFront to accept and forward these methods because you want to use the `PUT` method, you must configure Amazon S3 bucket policies or ACLs to handle `DELETE` requests appropriately so users can't delete resources that you don't want them to.

Note the following:

- You may find it easier to update Amazon S3 bucket policies than ACLs because you can add objects to the bucket without updating permissions. However, ACLs give you more fine-grained control because you're granting permissions on each object.
- By default, your Amazon S3 bucket and all of the objects in it are private—only the AWS account that created the bucket has permission to read or write the objects in it.
- If you're adding an origin access identity to an existing distribution, modify the bucket policy or any object ACLs as appropriate to ensure that the objects are not publicly available.
- Grant additional permissions to one or more secure administrator accounts so you can continue to update the contents of the Amazon S3 bucket.

Important

There may be a brief delay between when you save your changes to Amazon S3 permissions and when the changes take effect. Until the changes take effect, you may get permission-denied errors when you try to access objects in your bucket.

Updating Amazon S3 Bucket Policies

You can update the Amazon S3 bucket policy using either the AWS Management Console or the Amazon S3 API:

- Grant the CloudFront origin access identity the applicable permissions on the bucket.

  To specify an origin access identity, use the value of **Amazon S3 Canonical User ID** on the **Origin Access Identity** page in the CloudFront console. If you're using the CloudFront API, use the value of the `S3CanonicalUserId` element that was returned when you created the origin access identity.
- Deny access to anyone that you don't want to have access using Amazon S3 URLs.

For more information, go to Using Bucket Policies in the *Amazon Simple Storage Service Developer Guide*.

For an example, see "Granting Permission, Using Canonical ID, to a CloudFront Origin Identify" in the topic Example Cases for Amazon S3 Bucket Policies, also in the *Amazon Simple Storage Service Developer Guide*.

Updating Amazon S3 ACLs

Using either the AWS Management Console or the Amazon S3 API, change the Amazon S3 ACL:

- Grant the CloudFront origin access identity the applicable permissions on each object that the CloudFront distribution serves.

  To specify an origin access identity, use the value of **Amazon S3 Canonical User ID** on the **Origin Access Identity** page in the CloudFront console. If you're using the CloudFront API, use the value of the `S3CanonicalUserId` element that was returned when you created the origin access identity.
- Deny access to anyone that you don't want to have access using Amazon S3 URLs.

If another AWS account uploads objects to your bucket, that account is the owner of the objects. By default, the account that owns objects in a bucket is the only account that can grant permissions to those objects. However, the AWS account that owns the objects can make you an owner, too, which allows you to change permissions on the objects.

For more information, go to Using ACLs in the *Amazon Simple Storage Service Developer Guide*.

You can also change the ACLs using code and one of the AWS SDKs. For an example, see the downloadable sample code in Create a URL Signature Using C# and the .NET Framework (p. 157).

Specifying the AWS Accounts That Can Create Signed URLs (Trusted Signers)

Topics

- Creating CloudFront Key Pairs for Your Trusted Signers (p. 129)
- Reformatting the CloudFront Private Key (.NET and Java Only) (p. 129)
- Adding Trusted Signers to Your Distribution (p. 130)
- Verifying that Trusted Signers Are Active (Optional) (p. 132)
- Rotating CloudFront Key Pairs (p. 132)

To create signed URLs for the objects in your Amazon S3 bucket, you need at least one AWS account that has an active CloudFront key pair. This account is known as a trusted signer. The trusted signer has two purposes:

- As soon as you add the AWS account number for your trusted signer to your distribution, CloudFront starts to require that users use signed URLs to access the objects in your Amazon S3 bucket.
- When you create a signed URL, you use the private key from the trusted signer's key pair to sign a portion of the URL. When someone uses the signed URL to access an object, CloudFront compares the signed portion of the URL with the unsigned portion to verify that the URL hasn't been tampered with. CloudFront also verifies that the URL is valid, meaning, for example, that the expiration date and time in the URL hasn't passed.

When you specify trusted signers, you also indirectly specify the objects that require signed URLs:

- **Web distributions:** You add trusted signers to cache behaviors. If your distribution has only one cache behavior, users must use signed URLs to access any object associated with the distribution. If you create multiple cache behaviors and add trusted signers to some cache behaviors and not to others, you can require that users use signed URLs to access some objects and not others.
- **RTMP distributions:** You add trusted signers to a distribution. After you add trusted signers to an RTMP distribution, users must use signed URLs to access any object associated with the distribution.

> **Note**
> To specify trusted signers for a distribution, you must use the CloudFront console or CloudFront API version 2009-09-09 or later.

To specify the accounts that are allowed to create signed URLs and to add the accounts to your CloudFront distribution, perform the following tasks:

1. Decide which AWS accounts you want to use as trusted signers. Most CloudFront customers use the account that they used to created the distribution.
2. For each of the accounts that you selected in Step 1, create a CloudFront key pair. For more information, see Creating CloudFront Key Pairs for Your Trusted Signers (p. 129).
3. If you're using to .NET or Java to create signed URLs, reformat the CloudFront private key. For more information, see Reformatting the CloudFront Private Key (.NET and Java Only) (p. 129).
4. In the distribution for which you're creating signed URLs, specify the AWS account numbers of your trusted signers. For more information, see Adding Trusted Signers to Your Distribution (p. 130).
5. **Optional:** Verify that CloudFront recognizes that your trusted signers have active CloudFront key pairs. When trusted signers have active CloudFront key pairs, CloudFront recognizes them as active trusted signers. For more information, see Verifying that Trusted Signers Are Active (Optional) (p. 132).

Creating CloudFront Key Pairs for Your Trusted Signers

Each of the AWS accounts that you use to create CloudFront signed URLs—your trusted signers—must create its own CloudFront key pair, and the key pair must be active. Note the following:

- IAM users are currently not allowed to create CloudFront key pairs, so you cannot use IAM users as trusted signers.
- You cannot substitute an Amazon EC2 key pair for a CloudFront key pair. When you create a CloudFront signed URL, you include the key pair ID for the trusted signer's key pair in the URL. Amazon EC2 does not make key pair IDs available.

To help secure your applications, we recommend that you change CloudFront key pairs every 90 days or more often. For more information, see Rotating CloudFront Key Pairs (p. 132).

To create CloudFront key pairs for your trusted signers

1. Sign in to the AWS Management Console and open the Your Security Credentials page at https://console.aws.amazon.com/iam/home?#security_credential.
2. Expand **CloudFront Key Pairs**.
3. Click **Create New Key Pair**.
4. In the **Create Key Pair** dialog box, click **Download Private Key File**.
5. In the **Opening <filename>** dialog box, accept the default value of **Save File**, and click **OK** to download and save the private key for your CloudFront key pair.

   Important
   Save the private key for your CloudFront key pair in a secure location, and set permissions on the file so only the desired administrator users can read it. If someone gets your private key, they can generate valid signed URLs and download your content. You cannot get the private key again, so if you lose or delete it, you must create a new CloudFront key pair.
6. Record the key pair ID for your key pair. You'll use it when you create signed URLs.

Reformatting the CloudFront Private Key (.NET and Java Only)

If you're using .NET or Java to create signed URLs, you cannot use the private key from your key pair in the default .pem format to create the signature:

- **.NET framework:** Convert the private key to the XML format that the .NET framework uses. Several tools are available. For the example in Create a URL Signature Using C# and the .NET Framework (p. 157), we used .NET 2.0 OpenSSL Public and Private Key Parser, http://www.jensign.com/opensslkey/opensslkey.cs.
- **Java:** Convert the private key to DER format. To do this, you can use OpenSSL:

  ```
  $ openssl pkcs8 -topk8 -nocrypt -in origin.pem -inform PEM -out new.der -out-
  form DER
  ```

  To ensure that the encoder works correctly, add the jar for the Bouncy Castle Java cryptography APIs to your project and then add the Bouncy Castle provider.

Adding Trusted Signers to Your Distribution

Trusted signers are the AWS accounts that are allowed to create signed URLs for a distribution. By default, no account, not even the account that created the distribution, is allowed to create signed URLs for the distribution. To specify the AWS accounts that you want to use as trusted signers, add the accounts to your distribution:

- **Web distributions:** Trusted signers are associated with cache behaviors. This allows you to require signed URLs for some objects and not for others in the same distribution. Trusted signers can only create signed URLs for objects that are associated with the corresponding cache behaviors. For example, if you have one trusted signer for one cache behavior and a different trusted signer for a different cache behavior, neither trusted signer can create signed URLs for objects associated with the other cache behavior.
- **RTMP distributions:** Trusted signers are associated with the distribution. After you add trusted signers to an RTMP distribution, users must use signed URLs to access any of the objects associated with the distribution.

You must carefully define path patterns for your cache behaviors to ensure that CloudFront does or doesn't require signed URLs as appropriate.

> **Caution**
> Define path patterns and their sequence carefully to avoid providing end users with unintended access to your content. For example, suppose a request matches the path pattern for two cache behaviors. The first cache behavior does not require signed URLs and the second cache behavior does require signed URLs. End users will be able to access the objects without using a signed URL because CloudFront processes the cache behavior that is associated with the first match.

For more information about path patterns, see Path Pattern (p. 45).

> **Caution**
> If you're updating a distribution that you're already using to distribute content, add trusted signers only when you're ready to start generating signed URLs for your objects. After you add trusted signers to an RTMP distribution, users must use signed URLs to access any of the objects associated with the distribution. After you add trusted signers to a cache behavior for a web distribution, users must use signed URLs to access the objects associated with the cache behavior.

The maximum number of trusted signers depends on the type of distribution:

- **Web distributions:** A maximum of five for each cache behavior
- **RTMP distributions:** A maximum of five for the distribution

You can add trusted signers to your distribution using either the CloudFront console or the CloudFront API. See the applicable topic:

- Adding Trusted Signers to Your Distribution Using the CloudFront Console (p. 130)
- Adding Trusted Signers to Your Distribution Using the CloudFront API (p. 131)

Adding Trusted Signers to Your Distribution Using the CloudFront Console

To add trusted signers to your distribution using the CloudFront console

1. If you want to use only the AWS account that created the distribution as a trusted signer, skip to Step 2.

If you want to use other AWS accounts, get the AWS account number for each account:

a. Sign in to the AWS Management Console at https://console.aws.amazon.com/console/home using an account that you want to use as a trusted signer.
b. In the upper-right corner of the console, click the name associated with the account, and click **My Account**.
c. Make note of the account number that appears in the upper-right corner of the page.
d. Click **Sign Out**.
e. Repeat Step a through Step d for the other accounts that you want to use as trusted signers.

2. Open the Amazon CloudFront console at https://console.aws.amazon.com/cloudfront/, and sign in using the account that you used to create the distribution to which you want to add trusted signers.
3. Click the i icon for the distribution.
4. Change to edit mode:
   - **Web distributions:** Click the **Behaviors** tab, click the behavior that you want to edit, and click **Edit**.
   - **RTMP distributions:** Click **Edit**.
5. For **Restrict Viewer Access (Use Signed URLs)**, click **Yes**.
6. For **Trusted Signers**, check the applicable check boxes:
   - **Self:** Check this check box if you want to use the current account (the account that you used to create the distribution).
   - **Specify Accounts: Check this check box if you want to use other AWS accounts.**
7. If you checked the **Specify Accounts** check box, enter AWS account numbers (with or without hyphens) in the **AWS Account Number** field. Enter one account number per line.
8. Click **Yes, Edit**.
9. If you're adding trusted signers to a web distribution and you have more than one cache behavior, repeat Step 4 through Step 8 as applicable.

Adding Trusted Signers to Your Distribution Using the CloudFront API

You can use the CloudFront API to add the AWS account numbers for trusted signers to an existing distribution or to create a new distribution that includes trusted signers. In either case, specify the applicable values in the `TrustedSigners` element. For web distributions, add the `TrustedSigners` element to one or more cache behaviors. For RTMP distributions, add the `TrustedSigners` element to the distribution.

See the applicable topic in the *Amazon CloudFront API Reference*:

- **Create a new web distribution:** POST Distribution
- **Update an existing web distribution:** PUT Distribution Config
- **Create a new RTMP distribution:** POST Streaming Distribution
- **Update an existing RTMP distribution:** PUT Streaming Distribution Config

Verifying that Trusted Signers Are Active (Optional)

After you add trusted signers to your distribution, you may want to verify that the signers are active. For a trusted signer to be active, the following must be true:

- The AWS account must have at least one active key pair. If you're rotating key pairs, the account will temporarily have two active key pairs, the old key pair and the new one.
- CloudFront must be aware of the active key pair. After you create a key pair, there can be a short period of time before CloudFront is aware the key pair exists.

> **Note**
> To display a list of active trusted signers for a distribution, you currently must use the CloudFront API. A list of active trusted signers is not available in the CloudFront console.

Verifying that Trusted Signers Are Active Using the CloudFront API

To determine which trusted signers have active key pairs (are active trusted signers), you get the distribution and review the values in the `ActiveTrustedSigners` element. This element lists the AWS account number of each account that the distribution identifies as a trusted signer. If the trusted signer has one or more active CloudFront key pairs, the `ActiveTrustedSigners` element also lists the key pair IDs. For more information, see the applicable topic in the *Amazon CloudFront API Reference*:

- **Web distributions:** GET Distribution
- **RTMP distributions:** GET Streaming Distribution

Rotating CloudFront Key Pairs

AWS recommends that you rotate (change) your active CloudFront key pairs every 90 days. To rotate CloudFront key pairs that you are using to create signed URLs without invalidating URLs that haven't expired yet, perform the following tasks:

1. Create a new key pair for each of the accounts that you are using to create signed URLs. For more information, see Creating CloudFront Key Pairs for Your Trusted Signers (p. 129).
2. Verify that CloudFront is aware of the new key pairs. For more information, see Verifying that Trusted Signers Are Active (Optional) (p. 132).
3. Update your application to create signatures using the private keys from the new key pairs.
4. Confirm that signed URLs that you are signing using the new private keys are working.
5. Wait until the expiration date has passed in signed URLs that were signed using the old CloudFront key pairs.
6. Change the old CloudFront key pairs to **Inactive**:

   a. Go to the Your Security Credentials page.
   b. Expand **CloudFront Key Pairs**.
   c. For the applicable key pairs, click **Make Inactive**.

7. Reconfirm that signed URLs that you are signing using the new private keys are working.
8. Delete the old CloudFront key pairs:

   a. Go to the Your Security Credentials page.
   b. Expand **CloudFront Key Pairs**.
   c. For the applicable key pairs, click **Delete**.

9. Delete the old private keys from the location where you stored them.

Overview of Signed URLs

Topics

- Signed URLs that You Create Using a Canned Policy (p. 134)
- Signed URLs that You Create Using a Custom Policy (p. 134)
- The Parts of a Signed URL (p. 135)
- When Does CloudFront Check the Expiration Date and Time in a Signed URL? (p. 137)

Now that you've configured CloudFront to require that end users access your objects using special signed URLs, you can create the signed URLs (either manually or programmatically) and distribute them to the users that you want to have access to the objects.

A signed URL includes additional information, for example, an expiration date and time, that gives you more control over access to your content. This additional information appears in a policy statement, which is based on either a canned policy or a custom policy. The differences between canned and custom policies are explained in the next two sections.

> **Note**
> You can create some signed URLs using canned policies and create some signed URLs using custom policies for the same distribution.

Signed URLs that You Create Using a Canned Policy

Use a canned policy to create a CloudFront signed URL if you want to:

- Restrict access to a single object.
- Control access to your objects based only on the date and time that you want users to stop having access.

You don't include the policy in the URL, so a canned policy results in a shorter URL. You do include a signature, for which you hash and sign the policy.

When a user requests an object using a signed URL that you created using a canned policy, CloudFront reconstructs the canned policy statement based on information in the URL. CloudFront then compares the reconstructed policy statement with the policy statement in the signature to determine whether to allow the end user to access to the content. If the two statements don't match exactly, CloudFront denies access to the content.

For information about creating signed URLs using a canned policy, see Creating a Signed URL Using a Canned Policy (p. 138).

Signed URLs that You Create Using a Custom Policy

Use a custom policy to create a CloudFront signed URL if you want to:

- Restrict access to one or more objects, for example, all of the .pdf files in an annual-report directory. This allows you to use a single policy statement for the signed URLs for multiple objects.
- Control access to your objects based on:
  - The date and time that you want users to stop having access.

- The date and time that you want users to begin having access. (Optional)
- The IP address or range of IP addresses of the users who you want to have access to the object or objects. (Optional)

For a signed URL that you create using a custom policy, you hash and sign the policy and include the signature, as you do with a signed URL that you create using a canned policy. In addition, you include a Base64-encoded version of the policy in the URL, which provides URL-safe compression. As a result, using a custom policy results in a longer URL than a canned policy.

When a user requests an object using a signed URL that you created using a canned policy, CloudFront compares the custom policy statement in the signed URL with the policy statement in the signature to determine whether to allow the end user to access to the content. If the two statements don't match exactly, CloudFront denies access to the content.

For information about creating signed URLs using a custom policy, see Creating a Signed URL Using a Custom Policy (p. 143).

The Parts of a Signed URL

CloudFront signed URLs include the following components.

Base URL

The base URL is the CloudFront URL that you would use to access the object if you were not using signed URLs, including your own query string parameters, if any. For more information about the format of URLs for web distributions, see Format of URLs for CloudFront Objects (p. 72).

The following examples show values that you specify for web distributions.

- The following CloudFront URL is for an object in a web distribution (using the CloudFront domain name). Note that `image.jpg` is in an `images` directory. The path to the object in the URL must match the path to the object on your HTTP server or in your Amazon S3 bucket.

  ```
  http://d111111abcdef8.cloudfront.net/images/image.jpg
  ```
- The following CloudFront URL includes a query string:

  ```
  http://d111111abcdef8.cloudfront.net/images/image.jpg?size=large
  ```
- The following CloudFront URLs are for objects in a web distribution. Both use an alternate domain name; the second one includes a query string:

  ```
  http://www.example.com/images/image.jpg
  ```

  ```
  http://www.example.com/images/image.jpg?color=red
  ```
- The following CloudFront URL is for an objects in a web distribution that uses an alternate domain name and the HTTPS protocol:

  ```
  https://www.example.com/images/image.jpg
  ```

For RTMP distributions, the following examples are for objects in two different video formats, MP4 and FLV:

- **MP4:** `mp4:sydney-vacation.mp4`
- **FLV:** `sydney-vacation`
- **FLV:** `sydney-vacation.flv`

Note
For .flv files, whether you include the `.flv` filename extension depends on your player. To serve MP3 audio files or H.264/MPEG-4 video files, you might need to prefix the file name with `mp3:` or `mp4:`. Some media players can be configured to add the prefix automatically. The media player might also require you to specify the file name without the file extension (for example, sydney-vacation instead of sydney-vacation.mp4).

Expiration Date and Time (Canned Policies Only)

The date and time, in Unix time format (in seconds) and Coordinated Universal Time (UTC), that you want the URL to stop allowing access to the object. For example, January 1, 2013 10:00 am UTC converts to 1357034400 in Unix time format. For information about UTC, see *RFC 3339, Date and Time on the Internet: Timestamps*, http://tools.ietf.org/html/rfc3339.

If you're using a custom policy, you can still specify the date and time that you want a signed URL to stop allowing access to the object, but the date and time is included in the policy statement instead of appearing as a separate query string parameter in the signed URL.

Policy Statement

A policy statement is a text string in JSON format that determines the characteristics of the signed URL. For both canned policies and custom policies, the policy statement includes the URL of the object (for web distributions) or the stream name (for RTMP distributions), and an expiration date and time. Note the following:

- **Canned policy:** You don't include the policy statement in the signed URL—you only create a policy statement so you can hash and sign it, and include the signature in the URL. See Signature (p. 136).
- **Custom policy:** You can also optionally include a date and time that the URL becomes valid and/or an IP address or range of IP addresses that are allowed to access the object. Then you Base64-encode the policy statement, and you include the encoded policy statement in the signed URL.

Signature

The signature is a hashed and signed version of the policy statement.

CloudFront Key Pair ID for the AWS Account that Is Creating the Signed URL

The CloudFront key pair ID tells CloudFront which public key to use to validate the signed URL. CloudFront compares the information in the signature with the information in the policy statement to verify that the URL has not been tampered with.

The key pair ID that you include in CloudFront signed URLs must be the ID of an active key pair for one of your trusted signers. For more information, see Specifying the AWS Accounts That Can Create Signed URLs (Trusted Signers) (p. 128).

If you make a key pair inactive while rotating CloudFront key pairs, and if you're generating signed URLs programmatically, you must update your application to use a new active key pair for one of your trusted signers. If you're generating signed URLs manually, you must create new signed URLs. For more information about rotating key pairs, see Rotating CloudFront Key Pairs (p. 132).

When Does CloudFront Check the Expiration Date and Time in a Signed URL?

When CloudFront checks the expiration date and time in a signed URL to determine whether the URL is still valid depends on whether the URL is for a web distribution or an RTMP distribution:

- **Web distributions:** CloudFront checks the expiration date and time in a signed URL at the time of the HTTP request. If a client begins to download a large object immediately before the expiration time, the download should complete even if the expiration time passes during the download. If the TCP connection drops and the client tries to restart the download after the expiration time passes, the download will fail.

  If a client uses Range GETs to get an object in smaller pieces, any GET request that occurs after the expiration time passes will fail. For more information about Range GETs, see How CloudFront Processes Partial Requests for an Object (Range GETs) (p. 96).
- **RTMP distributions:** CloudFront checks the expiration time in a signed URL at the start of a play event. If a client starts to play a media file before the expiration time passes, CloudFront allows the entire media file to play. However, depending on the media player, pausing and restarting might trigger another play event. Skipping to another position in the media file will trigger another play event. If the subsequent play event occurs after the expiration time passes, CloudFront won't serve the media file.

Creating a Signed URL Using a Canned Policy

To create a signed URL using a canned policy

1. If you're using .NET or Java to create signed URLs, and if you haven't reformatted the private key for your key pair from the default .pem format to a format compatible with .NET or with Java, do so now. For more information, see Reformatting the CloudFront Private Key (.NET and Java Only) (p. 129).
2. Concatenate the following values in the specified order, and remove the whitespace between the parts. You might have to include escape characters in the string in application code. All values have a type of String. Each part is keyed by number (❶) to the two examples that follow.

❶	*`Base URL for the object`* This is the URL that you use to access the object if you aren't using a signed URL, for example: • **Web distribution:** `http://d111111abcdef8.cloudfront.net/images/image.jpg` • **RTMP distribution:** `videos/mediafile.flv`
❷	**?** The **?** indicates that query string parameters follow the base URL. Include the **?** even if you don't have any query string parameters of your own.
❸	*`Your query string parameters, if any`***&** This value is optional. If you want to add your own query string parameters, for example: `color=red&size=medium` then add the parameters after the **?** (see ❷) and before the `Expires` parameter. **Important** Your parameters cannot be named `Expires`, `Signature`, or `Key-Pair-Id`. If you add your own parameters, append an **&** after each one, including the last one.
❹	**Expires=***`date and time in Unix time format (in seconds) and Coordinated Universal Time (UTC)`* Specify the expiration date and time in Unix time format and Coordinated Universal Time (UTC). For example, January 1, 2013 10:00 am UTC converts to 1357034400 in Unix time format. For information about UTC, see *RFC 3339, Date and Time on the Internet: Timestamps*, http://tools.ietf.org/html/rfc3339.
❺	**&Signature=***`hashed and signed version of the policy statement`* A hashed and signed version of the policy statement. For more information, see Creating a Signature for a Canned Policy (p. 139).

❻	**&Key-Pair-Id=***`active CloudFront key pair Id for the key pair that you are using to generate the signature`* The ID for an active CloudFront key pair, for example, APKA9ONS7QCOWEXAMPLE: • **Web distributions:** The key pair must be associated with an AWS account that is one of the trusted signers for the applicable cache behavior. • **RTMP distributions:** The key pair must be associated with an AWS account that is one of the trusted signers for the distribution. For more information, see Specifying the AWS Accounts That Can Create Signed URLs (Trusted Signers) (p. 128).

Example signed URL for a web distribution:

❶http://d111111abcdef8.cloudfront.net/image.jpg ❷? ❸color=red&size=medium& ❹Expires=1357034400 ❺&Signature=nitfHRCrtziwO2HwPfWw~yYDhUF5EwRunQA-j19DzZr vDh6hQ73lDx~-ar3UocvvRQVw6EkC~GdpGQyyOSKQim-TxAnW7d8F5Kkai9HVx0FIu- 5jcQb0UEmatEXAMPLE3ReXySpLSMj0yCd3ZAB4UcBCAqEijkytL6f3fVYNGQI6 ❻&Key-Pair-Id=APKA9ONS7QCOWEXAMPLE

Example signed URL for an RTMP distribution:

❶videos/mediafile.flv ❷? ❸color=red&size=medium& ❹Expires=1357034400 ❺&Signature=nitfHRCrtziwO2HwPfWw~yYDhUF5EwRunQA-j19DzZr vDh6hQ73lDx~-ar3UocvvRQVw6EkC~GdpGQyyOSKQim-TxAnW7d8F5Kkai9HVx0FIu- 5jcQb0UEmatEXAMPLE3ReXySpLSMj0yCd3ZAB4UcBCAqEijkytL6f3fVYNGQI6 ❻&Key-Pair-Id=APKA9ONS7QCOWEXAMPLE

Creating a Signature for a Canned Policy

To create the signature for a signed URL that uses a canned policy, you perform two procedures:

- With the first procedure, immediately following, you create a policy statement.
- With the second procedure, you hash and sign the policy statement. There are two versions of this procedure. The version that you choose depends on your distribution type (web or RTMP) and, for RTMP distributions, the media player that you're using (Adobe Flash Player or another media player). Use the links after the first procedure to guide you to the applicable version of the second procedure.

For signed URLs that use a canned policy, you don't include the policy statement in the URL, as you do for signed URLs that use a custom policy.

For additional information and examples of how to hash, sign, and encode the policy statement, see:

- Using Linux Commands and OpenSSL for Base64-Encoding and Encryption (p. 151)
- Code and Examples for Creating a Signature for a Signed URL (p. 153)
- Tools for Configuring Private Content (p. 285)

To create the policy statement for a signed URL that uses a canned policy

1. Construct the policy statement using the following JSON format and using UTF-8 character encoding. Include all punctuation and other literal values exactly as specified.

   ```
   {"Statement":[{"Resource":"base URL or stream name","Condition":{"Date-
   LessThan":{"AWS:EpochTime":ending date and time in Unix time format and
   UTC}}}]}
   ```

   For Resource and DateLessThan, specify the following values:

   Resource: The value that you specify depends on whether you're creating the signed URL for a web distribution or an RTMP distribution:

   - **Web distributions:** The base URL including your query strings, if any, but excluding the CloudFront Expires, Signature, and Key-Pair-Id parameters, for example:

     ```
     http://d111111abcdef8.cloudfront.net/images/horizon.jpg?size=large&li-
     cense=yes
     ```

     Note the following:
     - The value must begin with http://, https://, or http*://.
     - If you have no query string parameters, omit the question mark.
     - If you specify an alternate domain name (CNAME) in the URL, you must specify the alternate domain name when referencing the object in your web page or application. Do not specify the Amazon S3 URL for the object.
   - **RTMP distributions:** Include only the stream name. For example, if the full URL for a streaming video is:

     ```
     rtmp://s5c39gqb8ow64r.cloudfront.net/videos/mp3_name.mp3
     ```

     then use the following value for Resource:

     ```
     videos/mp3_name
     ```

     Do not include a prefix such as mp3: or mp4:. Also, depending on the player you're using, you might have to omit the file extension from the value of Resource. For example, you might need to use sydney-vacation instead of sydney-vacation.flv.

   DateLessThan: The expiration date and time for the URL in Unix time format (in seconds) and Coordinated Universal Time (UTC). For example, January 1, 2013 10:00 am UTC converts to 1357034400 in Unix time format. For information about UTC, see *RFC 3339, Date and Time on the Internet: Timestamps*, http://tools.ietf.org/html/rfc3339.

   This value must match the value of the Expires query string parameter in the signed URL. Do not enclose the value in quotation marks.

   For more information, see When Does CloudFront Check the Expiration Date and Time in a Signed URL? (p. 137).

   Example

   When you use the following sample canned policy in a signed URL, an end user can access the object http://d111111abcdef8.cloudfront.net/horizon.jpg until January 1, 2013 10:00 am UTC:

```
{"Statement":[{"Resource":"http://d111111abcdef8.cloudfront.net/hori
zon.jpg?size=large&license=yes","Condition":{"DateLessThan":{"AWS:Epoch
Time":1357034400}}}]}
```

If you copy and paste this example, replace the URL and expiration time with your own values.

2. Remove any whitespace from the policy statement. You might have to include escape characters in the string in application code.

Perform the applicable procedure to create the signature for your signed URL:

- Option 1: To create a signature for a web distribution or for an RTMP distribution (without Adobe Flash Player) by using a canned policy (p. 141)
- Option 2: To create a signature for an RTMP distribution by using a canned policy (Adobe Flash Player) (p. 141)

Option 1: To create a signature for a web distribution or for an RTMP distribution (without Adobe Flash Player) by using a canned policy

1. Use the SHA-1 hash function to hash and sign the policy statement that you created in the To create the policy statement for a signed URL that uses a canned policy (p. 140) procedure. For the private key that is required by the hash function, use the private key that is associated with the applicable active trusted signer.

   Note
   The method that you use to hash and sign the policy statement depends on your programming language and platform. For sample code, see Code and Examples for Creating a Signature for a Signed URL (p. 153).

2. Remove whitespace from the hashed and signed string.
3. Base64-encode the string.
4. Replace characters that are invalid in a URL query string with characters that are valid. The following table lists invalid and valid characters.

Replace these invalid characters	With these valid characters
+	- (hyphen)
=	_ (underscore)
/	~ (tilde)

5. Append the resulting value to your signed URL after **&Signature=**, and return to To create a signed URL using a canned policy (p. 138) to finish concatenating the parts of your signed URL.

Option 2: To create a signature for an RTMP distribution by using a canned policy (Adobe Flash Player)

1. Use the SHA-1 hash function to hash and sign the policy statement that you created in the To create the policy statement for a signed URL that uses a canned policy (p. 140) procedure. For the private key that is required by the hash function, use the private key that is associated with the applicable active trusted signer.

Note

The method that you use to hash and sign the policy statement depends on your programming language and platform. For sample code, see Code and Examples for Creating a Signature for a Signed URL (p. 153).

2. Remove whitespace from the hashed and signed string.

   Continue on to Step 3 if you're using Adobe Flash Player and the stream name is passed in from a web page.

   If you're using Adobe Flash Player and if the stream name is not passed in from a web page, skip the rest of this procedure. For example, if you wrote your own player that fetches stream names from within the Adobe Flash .swf file, skip the rest of this procedure.
3. Base64-encode the string.
4. Replace characters that are invalid in a URL query string with characters that are valid. The following table lists invalid and valid characters.

Replace these invalid characters	With these valid characters
+	- (hyphen)
=	_ (underscore)
/	~ (tilde)

5. Some versions of Adobe Flash Player require that you URL-encode the characters ?, =, and &. For information about whether your version of Adobe Flash Player requires this character substitution, refer to the Adobe website.

   If your version of Flash does not require URL-encoding those character, skip to Step 6.

   If your version of Flash requires URL-encoding those characters, replace them as indicated in the following table. (You already replaced = in the previous step.)

Replace these invalid characters	With this URL encoding
?	%3F
&	%26

6. Append the resulting value to your signed URL after **&Signature=**, and return to To create a signed URL using a canned policy (p. 138) to finish concatenating the parts of your signed URL.

Creating a Signed URL Using a Custom Policy

Topics

- Creating a Policy Statement for a Custom Policy (p. 144)
- Example Policy Statements for a Custom Policy (p. 146)
- Creating a Signature for a Custom Policy (p. 148)

To create a signed URL using a custom policy, perform the following procedure.

To create a signed URL using a custom policy

1. If you're using .NET or Java to create signed URLs, and if you haven't reformatted the private key for your key pair from the default .pem format to a format compatible with .NET or with Java, do so now. For more information, see Reformatting the CloudFront Private Key (.NET and Java Only) (p. 129).
2. Concatenate the following values in the specified order, and remove the whitespace between the parts. You might have to include escape characters in the string in application code. All values have a type of String. Each part is keyed by number (❶) to the two examples that follow.

❶	*Base URL for the object* This is the URL that you use to access the object if you aren't using a signed URL, for example: • **Web distribution:** `http://d111111abcdef8.cloudfront.net/images/image.jpg` • **RTMP distribution:** `videos/mediafile.flv`
❷	**?** The **?** indicates that query string parameters follow the base URL. Include the **?** even if you don't have any query string parameters of your own.
❸	*Your query string parameters, if any***&** This value is optional. If you want to add your own query string parameters, for example: `color=red&size=medium` then add them after the **?** (see ❷) and before the `Policy` parameter. **Important** Your parameters cannot be named `Policy`, `Signature`, or `Key-Pair-Id`. If you add your own parameters, append an **&** after each one, including the last one.
❹	**Policy=***policy statement* Your policy statement in JSON format, with white space removed. For more information, see Creating a Policy Statement for a Custom Policy (p. 144).
❺	**&Signature=***hashed and signed version of the policy statement* A hashed and signed version of the policy statement. For more information, see Creating a Signature for a Custom Policy (p. 148).

❻	**&Key-Pair-Id=***active CloudFront key pair Id for the key pair that you are using to sign the policy statement* The ID for an active CloudFront key pair, for example, APKA9ONS7QCOWEXAMPLE: • **Web distributions:** The key pair must be associated with an AWS account that is one of the trusted signers for the applicable cache behavior. • **RTMP distributions:** The key pair must be associated with an AWS account that is one of the trusted signers for the distribution. For more information, see Specifying the AWS Accounts That Can Create Signed URLs (Trusted Signers) (p. 128).

Example signed URL for a web distribution:

❶http://d111111abcdef8.cloudfront.net/image.jpg ❷? ❸color=red&size=medium& ❹Policy=eyANCiAgICEXAMPLEW1lbnQiOiBbeyANCiAgICAgICJSZXNvdXJjZSI6Imh0dHA 6Ly9kemJlc3FtN3VuMW0wLmNsb3VkZnJvbnQubmV0L2RlbW8ucGhwIiwgDQogICAgICAiQ 29uZGl0aW9uIjp7IA0KICAgICAgICAgIklwQWRkcmVzcyI6eyJBV1M6U291cmNlSXAiOiI yMDcuMTcxLjE4MC4xMDEvMzIifSwNCiAgICAgICAgICJEYXRlR3JlYXRlclRoYW4iOnsiQ VdTOkVwb2NoVGltZSI6MTI5Njg2MDE3Nn0sDQogICAgICAgICAiRGF0ZUxlc3NUaGFuIjp 7IkFXUzpFcG9jaFRpbWUiOjEyOTY4NjAyMjZ9DQogICAgICB9IA0KICAgfV0gDQp9DQo ❺&Signature=nitfHRCrtziwO2HwPfWw~yYDhUF5EwRunQA-j19DzZrvDh6hQ73lDx~ -ar3UocvvRQVw6EkC~GdpGQyyOSKQim-TxAnW7d8F5Kkai9HVx0FIu-5jcQb0UEmat EXAMPLE3ReXySpLSMj0yCd3ZAB4UcBCAqEijkytL6f3fVYNGQI6 ❻&Key-Pair-Id=APKA9ONS7QCOWEXAMPLE

Example signed URL for an RTMP distribution:

❶videos/mediafile.flv ❷? ❸color=red&size=medium& ❹Policy=eyANCiAgICEXAMPLEW1lbnQiOiBbeyANCiAgICAgICJSZXNvdXJjZSI6Imh0dHA 6Ly9kemJlc3FtN3VuMW0wLmNsb3VkZnJvbnQubmV0L2RlbW8ucGhwIiwgDQogICAgICAiQ 29uZGl0aW9uIjp7IA0KICAgICAgICAgIklwQWRkcmVzcyI6eyJBV1M6U291cmNlSXAiOiI yMDcuMTcxLjE4MC4xMDEvMzIifSwNCiAgICAgICAgICJEYXRlR3JlYXRlclRoYW4iOnsiQ VdTOkVwb2NoVGltZSI6MTI5Njg2MDE3Nn0sDQogICAgICAgICAiRGF0ZUxlc3NUaGFuIjp 7IkFXUzpFcG9jaFRpbWUiOjEyOTY4NjAyMjZ9DQogICAgICB9IA0KICAgfV0gDQp9DQo ❺&Signature=nitfHRCrtziwO2HwPfWw~yYDhUF5EwRunQA-j19DzZrvDh6hQ73lDx~ -ar3UocvvRQVw6EkC~GdpGQyyOSKQim-TxAnW7d8F5Kkai9HVx0FIu-5jcQb0UEmat EXAMPLE3ReXySpLSMj0yCd3ZAB4UcBCAqEijkytL6f3fVYNGQI6 ❻&Key-Pair-Id=APKA9ONS7QCOWEXAMPLE

Creating a Policy Statement for a Custom Policy

To create a policy statement for a custom policy, perform the following procedure. For several example policy statements that control access to objects in a variety of ways, see Example Policy Statements for a Custom Policy (p. 146).

To create the policy statement for a signed URL that uses a custom policy

1. Construct the policy statement using the following JSON format.

```
{
   "Statement": [{
      "Resource":"URL or stream name of the object",
      "Condition":{
         "DateLessThan":{"AWS:EpochTime":required ending date and time in
Unix time format and UTC},
        "DateGreaterThan":{"AWS:EpochTime":optional beginning date and time
 in Unix time format and UTC},
         "IpAddress":{"AWS:SourceIp":"optional IP address"}
      }
   }]
}
```

Note the following:

- Use UTF-8 character encoding.
- Include all punctuation and parameter names exactly as specified. Abbreviations for parameter names are not accepted.
- The order of the parameters in the `Condition` section doesn't matter.
- For information about the values for `Resource`, `DateLessThan`, `DateGreaterThan`, and `IpAddress`, see the descriptions after this procedure.

2. Remove any whitespace from the policy statement. You might have to include escape characters in the string in application code.
3. Append the resulting value to your signed URL after `Policy=`.
4. Create a signature for the signed URL by hashing, signing, and Base64-encoding the policy statement. For more information, see Creating a Signature for a Custom Policy (p. 148).

Resource

- **Web distributions (optional but recommended):** The base URL including your query strings, if any, but excluding the CloudFront `Policy`, `Signature`, and `Key-Pair-Id` parameters, for example:

  ```
  http://d111111abcdef8.cloudfront.net/images/horizon.jpg?size=large&license=yes
  ```

  Caution
  If you omit the Resource parameter for a web distribution, end users can access all of the objects associated with any distribution that is associated with the key pair that you use to create the signed URL.

  Note the following:

  - The value must begin with `http://`, `https://`, or `*`.
  - If you have no query string parameters, omit the question mark.
  - You can use the wild card character that matches zero or more characters (*) or the wild-card character that matches exactly one character (?) anywhere in the string. For example, the value:

    ```
    http*://d111111abcdef8.cloudfront.net/*game_download.zip*
    ```

    would include (for example), the following objects:

    ```
    http://d111111abcdef8.cloudfront.net/example_game_download.zip?license=yes
    ```

    ```
    https://d111111abcdef8.cloudfront.net/example_game_download.zip?license=yes
    ```

```
http://d111111abcdef8.cloudfront.net/test_game_download.zip?license=temp
```

```
https://d111111abcdef8.cloudfront.net/test_game_download.zip?license=temp
```

- If you specify an alternate domain name (CNAME) in the URL, you must specify the alternate domain name when referencing the object in your web page or application. Do not specify the Amazon S3 URL for the object.

- **RTMP distributions:** Include only the stream name. For example, if the full URL for a streaming video is:

  ```
  rtmp://s5c39gqb8ow64r.cloudfront.net/videos/mp3_name.mp3
  ```

  then use the following value for `Resource`:

  ```
  videos/mp3_name
  ```

  Do not include a prefix such as `mp3:` or `mp4:`. Also, depending on the player you're using, you might have to omit the file extension from the value of `Resource`. For example, you might need to use `sydney-vacation` instead of `sydney-vacation.flv`.

DateLessThan

The expiration date and time for the URL in Unix time format (in seconds) and Coordinated Universal Time (UTC). Do not enclose the value in quotation marks. For information about UTC, see *RFC 3339, Date and Time on the Internet: Timestamps*, http://tools.ietf.org/html/rfc3339.

For example, January 1, 2013 10:00 am UTC converts to 1357034400 in Unix time format.

This is the only required parameter in the `Condition` section. CloudFront requires this value to prevent users from having permanent access to your private content.

For more information, see When Does CloudFront Check the Expiration Date and Time in a Signed URL? (p. 137)

DateGreaterThan (Optional)

An optional start date and time for the URL in Unix time format (in seconds) and Coordinated Universal Time (UTC). Users are not allowed to access the object before the specified date and time. Do not enclose the value in quotation marks.

IpAddress (Optional)

The IP address of the client making the GET request. To allow any IP address to access the object, omit this parameter.

IP address ranges must be in standard IPv4 CIDR format (for example, `10.52.176.0/24`). For more information, go to *RFC 4632, Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan*, http://tools.ietf.org/html/rfc4632.

You can specify only a single value for the condition. For example, you can't set the policy to allow access if the client's IP address is in one of two separate ranges.

Example Policy Statements for a Custom Policy

The following example policy statements show how to control access to a specific object, all of the objects in a directory, or all of the objects associated with a key pair ID. The examples also show how to control access from an individual IP address or a range of IP addresses, and how to prevent users from using the signed URL after a specified date and time.

If you copy and paste any of these examples, remove any whitespace, replace the applicable values with your own values, and include a newline character after the closing brace (}).

Example Policy Statement: Accessing One Object from a Range of IP Addresses

The following example custom policy in a signed URL specifies that an end user can access the object `http://d111111abcdef8.cloudfront.net/game_download.zip` from IP addresses in the range `192.0.2.0/24` until January 1, 2013 10:00 am UTC:

```
{
   "Statement": [{
      "Resource":"http://d111111abcdef8.cloudfront.net/game_download.zip",
      "Condition":{
         "IpAddress":{"AWS:SourceIp":"192.0.2.0/24"},
         "DateLessThan":{"AWS:EpochTime":1357034400}
      }
   }]
}
```

Example Policy Statement: Accessing All Objects in a Directory from a Range of IP Addresses

The following example custom policy allows you to create signed URLs for any object in the `training` directory, as indicated by the * wildcard character in the `Resource` parameter. End users can access the object from an IP address in the range `192.0.2.0/24` until January 1, 2013 10:00 am UTC:

```
{
   "Statement": [{
      "Resource":"http://d111111abcdef8.cloudfront.net/training/*",
      "Condition":{
         "IpAddress":{"AWS:SourceIp":"192.0.2.0/24"},
         "DateLessThan":{"AWS:EpochTime":1357034400}
      }
   }]
}
```

Each signed URL in which you use this policy includes a base URL that identifies a specific object, for example:

```
http://d111111abcdef8.cloudfront.net/training/orientation.pdf
```

Example Policy Statement: Accessing All Objects Associated with a Key Pair ID from One IP Address

The following sample custom policy allows you to create signed URLs for any object associated with any distribution, as indicated by the * wildcard character in the `Resource` parameter. The end user must use the IP address `192.0.2.10/32`. (The value `192.0.2.10/32` in CIDR notation refers to a single IP address, `192.0.2.10`.) The objects are available only from January 1, 2013 10:00 am UTC until January 2, 2013 10:00 am UTC:

```
{
   "Statement": [{
      "Resource":"http://*",
      "Condition":{
         "IpAddress":{"AWS:SourceIp":"192.0.2.10/32"},
         "DateGreaterThan":{"AWS:EpochTime":1357034400},
         "DateLessThan":{"AWS:EpochTime":1357120800}
      }
   }]
}
```

Each signed URL in which you use this policy includes a base URL that identifies a specific object in a specific CloudFront distribution, for example:

```
http://d111111abcdef8.cloudfront.net/training/orientation.pdf
```

The signed URL also includes a key pair ID, which must be associated with a trusted signer in the distribution (d111111abcdef8.cloudfront.net) that you specify in the base URL.

Creating a Signature for a Custom Policy

The signature for a signed URL that uses a custom policy is a hashed, signed, and Base64-encoded version of the policy statement. To create a signature for a custom policy, perform the applicable procedure. The version that you choose depends on your distribution type (web or RTMP) and, for RTMP distributions, the media player that you're using (Adobe Flash Player or another media player):

- Option 1: To create a signature for a web distribution or for an RTMP distribution (without Adobe Flash Player) by using a custom policy (p. 148)
- Option 2: To create a signature for an RTMP distribution by using a custom policy (Adobe Flash Player) (p. 149)

For additional information and examples of how to hash, sign, and encode the policy statement, see:

- Using Linux Commands and OpenSSL for Base64-Encoding and Encryption (p. 151)
- Code and Examples for Creating a Signature for a Signed URL (p. 153)
- Tools for Configuring Private Content (p. 285)

Option 1: To create a signature for a web distribution or for an RTMP distribution (without Adobe Flash Player) by using a custom policy

1. Use the SHA-1 hash function to hash and sign the policy statement that you created in the To create the policy statement for a signed URL that uses a custom policy (p. 144) procedure. For the private key that is required by the hash function, use the private key that is associated with the applicable active trusted signer.

   Note
   The method that you use to hash and sign the policy statement depends on your programming language and platform. For sample code, see Code and Examples for Creating a Signature for a Signed URL (p. 153).

2. Remove whitespace from the hashed and signed string.
3. Base64-encode the string.
4. Replace characters that are invalid in a URL query string with characters that are valid. The following table lists invalid and valid characters.

Replace these invalid characters	With these valid characters
+	- (hyphen)
=	_ (underscore)
/	~ (tilde)

5. Append the resulting value to your signed URL after **&Signature=**, and return to To create a signed URL using a custom policy (p. 143) to finish concatenating the parts of your signed URL.

Option 2: To create a signature for an RTMP distribution by using a custom policy (Adobe Flash Player)

1. Use the SHA-1 hash function to hash and sign the policy statement that you created in the To create the policy statement for a signed URL that uses a custom policy (p. 144) procedure. For the private key that is required by the hash function, use the private key that is associated with the applicable active trusted signer.

   Note
   The method that you use to hash and sign the policy statement depends on your programming language and platform. For sample code, see Code and Examples for Creating a Signature for a Signed URL (p. 153).

2. Remove whitespace from the hashed and signed string.

   Continue on to Step 3 if the stream name is passed in from a web page.

   If the stream name is not passed in from a web page, skip the rest of this procedure. For example, if you wrote your own player that fetches stream names from within the Adobe Flash .swf file, skip the rest of this procedure.

3. Base64-encode the string.
4. Replace characters that are invalid in a URL query string with characters that are valid. The following table lists invalid and valid characters.

Replace these invalid characters	With these valid characters
+	- (hyphen)
=	_ (underscore)
/	~ (tilde)

5. Some versions of Adobe Flash Player require that you URL-encode the characters ?, =, and &. For information about whether your version of Adobe Flash Player requires this character substitution, refer to the Adobe website.

   If your version of Adobe Flash Player does not require that you URL-encode the characters ?, =, and &, skip to Step 6.

   If your version of Adobe Flash Player requires URL-encoding those characters, replace them as indicated in the following table. (You already replaced = in the previous step.)

Replace these invalid characters	With this URL encoding
?	%3F

Replace these invalid characters	With this URL encoding
&	%26

6. Append the resulting value to your signed URL after **&Signature=**, and return to To create a signed URL using a custom policy (p. 143) to finish concatenating the parts of your signed URL.

Using Linux Commands and OpenSSL for Base64-Encoding and Encryption

You can use Linux command-line commands and OpenSSL to:

- Base64-encode the policy statement and replace invalid characters with valid characters.
- Convert the policy statement into a signature.

For information about OpenSSL, go to http://www.openssl.org.

Base64-Encoding the Policy Statement

The following Linux command Base64-encodes the policy statement (in the file policy) and replaces characters that are not valid in URL query string parameters with characters that are valid:

```
❶ cat policy | ❷ openssl base64 | ❸ tr '+=/' '-_~'
```

where:

❶	`cat` sends the policy file to openssl.
❷	OpenSSL Base64-encodes the file.
❸	`tr` replaces characters that are not valid in URL query string parameters with characters that are valid.

Converting the Policy Statement into a Signature

The following Linux command hashes, signs, and Base64-encodes the policy statement to create a signature:

```
❶ cat policy | ❷ openssl sha1 -sign private-key.pem | ❸ openssl base64
| ❹ tr '+=/' '-_~'
```

where:

❶	`cat` sends the Base64-encoded `policy` file to OpenSSL.
❷	OpenSSL hashes the file using SHA-1 and signs it using the private key file `private-key.pem`.
❸	OpenSSL Base64-encodes the hashed and signed policy statement.
❹	`tr` replaces characters that are not valid in URL query string parameters with characters that are valid.

Note
Remove whitespace, if any, from the resulting signature.

For code examples that demonstrate creating a signature in several programming languages see Code and Examples for Creating a Signature for a Signed URL (p. 153).

Code and Examples for Creating a Signature for a Signed URL

This section includes downloadable application samples that demonstrate how to create signatures for signed URLs. Examples are available in Perl, PHP, C#, and Java. All the samples can be used to create signed URLs. The Perl script runs on Linux/Mac platforms. The PHP example will work on any server that runs PHP. The C# example uses the .NET Framework.

Topics

- Create a URL Signature Using Perl (p. 153)
- Create a URL Signature Using PHP (p. 154)
- Create a URL Signature Using C# and the .NET Framework (p. 157)
- Create a URL Signature Using Java (p. 165)

Create a URL Signature Using Perl

The Perl script creates the signature for private content from command line arguments that specify the CloudFront URL, the path to the private key of the signer, the key ID, and an expiration date for the URL. The tool can also decode signed URLs. To get the tool `cfsign.pl`, go to Amazon CloudFront Signed URLs Helper Tool.

Note
Creating a URL signature is just one part of the process of serving private content using a signed URL. For more information about the entire process, see How Private Content Works (p. 119).

The following example shows how you might use `cfsign.pl` to create an RTMP distribution signature.

```
$ cfsign.pl --action encode --stream example/video.mp4 --private-key
  /path/to/my-private-key.pem --key-pair-id PK12345EXAMPLE --expires 1265838202
```

This tool generates the policy statement from the command line arguments. The signature that it generates is an SHA1 hash of the policy statement.

The following is a sample Base64-encoded stream name.

```
mp4:example/video.mp4%3FPolicy%3DewogICJTdGF0ZW1lbnQiOlt7CiAgICAgICJSZXNvdXJjZSI
6ImRyciIsCiAgICAgICJDb25kaXRpb24iOnsKICAgICAgICAiSXBBZGRyZXNzIjp7IkFXUzpTb3VyY2V
JcCI6IjAuMC4wLjAvMCJ9LAogICAgICAgICJEYXRlTGVzc1RoYW4iOnsiQVdTOkVwb2NoVGltZSI6MjE
0NTkxNjgwMH0KICAgICAgfQogICAgEXAMPLE_%26Signature%3DewtHqEXK~68tsZt-eOFnZKGwTf2a
JlbKhXkK5SSiVqcG9pieCRV3xTEPtc29OzeXlsDvRycOM2WK0cXzcyYZhpl9tv2796ihHiCTAwIHQ8yP
17Af4nWtOLIZHoH6wkR3tU1cQHs8R1d-g-SlZGjNBXr~J2MbaJzm8i6EXAMPLE_%26Key-Pair-Id%3
DPK12345EXAMPLE
```

This signature authenticates the request to stream private content, `example/video.mp4`. If you're using Adobe Flash Player and the stream name is passed in from a web page using JavaScript, you must Base64-encode the signature and replace characters that are invalid in a URL request parameter (+, =, /) with characters that are valid (-, _, and ~, respectively). If the stream name is not passed in from a web page, you don't need to Base64-encode the signature. For example, you would not Base64-encode the signature if you write your own player and the stream names are fetched from within the Adobe Flash .swf file.

The following example uses jwplayer with CloudFront.

```
<script type='text/javascript'>
  var so1 = new SWFObject
    ('http://d84l721fxaaqy9.cloudfront.net/player/player.swf',
    'mpl', '640', '360', '9');
  so1.addParam('allowfullscreen','true');
  so1.addParam('allowscriptaccess','always');
  so1.addParam('wmode','opaque');
  so1.addVariable('streamer','rtmp://s33r3xe4ayhhis.cloudfront.net/cfx/st');
  so1.addVariable("file","mp4:example/video.mp4%3FPolicy%3DewogICJTdGF0ZW1lbnQi

    Olt7CiAgICAgICJSZXNvdXJjZSI6ImRyciIsCiAgICAgICJDb25kaXRpb24iOnsKICAgICAgICA

    iSXBBZGRyZXNzIjp7IkFXUzpTb3VyY2VJcCI6IjAuMC4wLjAvMCJ9LAogICAgICAgICJEYXRlTG

    Vzc1RoYW4iOnsiQVdTOkVwb2NoVGltZSI6MjE0NTkxNjgwMH0KICAgICAgfQogICAgEXAMPLE_%

    26Signature%3DewtHqEXK~68tsZt-eOFnZKGwTf2aJlbKhXkK5SSiVqcG9pieCRV3xTEPtc29O

    zeXlsDvRycOM2WK0cXzcyYZhpl9tv2796ihHiCTAwIHQ8yP17Af4nWtOLIZHoH6wkR3tU1cQHs8

     R1d-g-SlZGjNBXr~J2MbaJzm8i6EXAMPLE_%26Key-Pair-Id%3DPK12345EXAMPLE
  so1.write('flv');
</script>
```

When you retrieve a stream to play from within an Adobe Flash .swf file, do not URL-encode the stream name, for example:

```
mp4:example/video.mp4?Policy=ewogICJTdGF0ZW1lbnQiOlt7CiAgICAgICJSZXNvdXJjZSI6ImR
yciIsCiAgICAgICJDb25kaXRpb24iOnsKICAgICAgICAiSXBBZGRyZXNzIjp7IkFXUzpTb3VyY2VJcCI
6IjAuMC4wLjAvMCJ9LAogICAgICAgICJEYXRlTGVzc1RoYW4iOnsiQVdTOkVwb2NoVGltZSI6MjE0NTk
xNjgwMH0KICAgICAgfQogICAgEXAMPLE_&Signature=ewtHqEXK~68tsZt-eOFnZKGwTf2aJlbKhXkK
5SSiVqcG9pieCRV3xTEPtc29OzeXlsDvRycOM2WK0cXzcyYZhpl9tv2796ihHiCTAwIHQ8yP17Af4nWt
OLIZHoH6wkR3tU1cQHs8R1d-g-SlZGjNBXr~J2MbaJzm8i6EXAMPLE_&Key-Pair-Id=PK12345
EXAMPLE
```

See the comments in the Perl source code for more information about the command line switches and features of this tool.

See also

- Create a URL Signature Using PHP (p. 154)
- Create a URL Signature Using C# and the .NET Framework (p. 157)
- Create a URL Signature Using Java (p. 165)
- Tools for Configuring Private Content (p. 285)

Create a URL Signature Using PHP

Any web server that runs PHP can use the PHP demo code to create policy statements and signatures for private CloudFront RTMP distributions. The sample creates a functioning web page with signed URL links that play a video stream using CloudFront streaming. To get the sample, download Signature Code for Video Streaming in PHP.

> **Note**
> Creating a URL signature is just one part of the process of serving private content using a signed URL. For more information about the entire process, see How Private Content Works (p. 119).

In the following code segment, the function rsa_sha1_sign hashes and signs the policy statement. The arguments required are a policy statement, an out parameter to contain the signature, and the private key for your AWS account or for a trusted AWS account that you specify. Next, the url_safe_base64_encode function creates a URL-safe version of the signature.

Example RSA SHA1 Hashing in PHP

```
 function rsa_sha1_sign($policy, $private_key_filename) {
    $signature = "";

    // load the private key
    $fp = fopen($private_key_filename, "r");
    $priv_key = fread($fp, 8192);
    fclose($fp);
    $pkeyid = openssl_get_privatekey($priv_key);

    // compute signature
    openssl_sign($policy, $signature, $pkeyid);

    // free the key from memory
    openssl_free_key($pkeyid);

    return $signature;
 }

function url_safe_base64_encode($value) {
    $encoded = base64_encode($value);
    // replace unsafe characters +, = and / with
    // the safe characters -, _ and ~
    return str_replace(
        array('+', '=', '/'),
        array('-', '_', '~'),
        $encoded);
 }
```

The following code constructs a *canned* policy statement needed for creating the signature. For more information about canned policies, see Creating a Signed URL Using a Canned Policy (p. 138).

Example Canned Signing Function in PHP

```
function get_canned_policy_stream_name($video_path, $private_key_filename, 
$key_pair_id, $expires) {
    // this policy is well known by CloudFront, but you still need to sign it,

    // since it contains your parameters
    $canned_policy = '{"Statement":[{"Resource":"' . $video_path . '","Condi
tion":{"DateLessThan":{"AWS:EpochTime":'. $expires . '}}}]}';
    // the policy contains characters that cannot be part of a URL, 
    // so we Base64 encode it
    $encoded_policy = url_safe_base64_encode($canned_policy);
    // sign the original policy, not the encoded version
    $signature = rsa_sha1_sign($canned_policy, $private_key_filename);
    // make the signature safe to be included in a url
    $encoded_signature = url_safe_base64_encode($signature);

    // combine the above into a stream name
    $stream_name = create_stream_name($video_path, null, $encoded_signature, 
$key_pair_id, $expires);
    // url-encode the query string characters to work around a flash player bug

    return encode_query_params($stream_name);
    }
```

The following code constructs a *custom* policy statement needed for creating the signature. For more information about custom policies, see Creating a Signed URL Using a Custom Policy (p. 143).

Example Custom Signing Function in PHP

```
function get_custom_policy_stream_name($video_path, $private_key_filename, 
$key_pair_id, $policy) {
    // the policy contains characters that cannot be part of a URL, 
    // so we Base64 encode it
    $encoded_policy = url_safe_base64_encode($policy);
    // sign the original policy, not the encoded version
    $signature = rsa_sha1_sign($policy, $private_key_filename);
    // make the signature safe to be included in a url
    $encoded_signature = url_safe_base64_encode($signature);

    // combine the above into a stream name
    $stream_name = create_stream_name($video_path, $encoded_policy, $encoded_sig
nature, $key_pair_id, null);
    // url-encode the query string characters to work around a flash player bug

    return encode_query_params($stream_name);

    }
```

For more information about the OpenSSL implementation of SHA-1, see The Open Source Toolkit for SSL/TLS.

See also

- Create a URL Signature Using Perl (p. 153)
- Create a URL Signature Using C# and the .NET Framework (p. 157)
- Create a URL Signature Using Java (p. 165)

- Tools for Configuring Private Content (p. 285)

Create a URL Signature Using C# and the .NET Framework

The C# examples in this section implement a sample application that demonstrates how to create the signatures for CloudFront private distributions using canned and custom policy statements. The samples includes utility functions based on the AWS .NET SDK that can be useful in .NET applications.

> **Note**
> Creating a URL signature is just one part of the process of serving private content using a signed URL. For more information about the entire process, see How Private Content Works (p. 119).

To download the code, go to Signature Code in C#.

To use the RSA keys provided by AWS Account/Security in the .NET framework, you must convert the AWS-supplied .pem files to the XML format that the .NET framework uses. The OpenSSL Public and Private Key Parser available from .NET 2.0 OpenSSL Public and Private Key Parser will do the conversion.

After conversion, the RSA private key file is in the following format:

Example RSA Private Key in the XML .NET Framework Format

```
<RSAKeyValue>
  <Modulus>
    wO5IvYCP5UcoCKDo1dcspoMehWBZcyfs9QEzGi6Oe5y+ewGr1oW+vB2GPB
    ANBiVPcUHTFWhwaIBd3oglmF0lGQljP/jOfmXHUK2kUUnLnJp+oOBL2Ni
uFtqcW6h/L5lIpD8Yq+NRHg
    Ty4zDsyr2880MvXv88yEFURCkqEXAMPLE=
  </Modulus>
  <Exponent>AQAB</Exponent>
  <P>
    5bmKDaTz
    npENGVqz4Cea8XPH+sxt+2VaAwYnsarVUoS
BeVt8WLloVuZGG9IZYmH5KteXEu7fZveYd9UEXAMPLE==
  </P>
  <Q>
    1v9l/WN1a1N3rOK4VGoCokx7kR2SyTMSbZgF9IWJNOugR/WZw7HTnjipO3c9dy1Ms9pUKwUF4
    6d7049EXAMPLE==
  </Q>
  <DP>
    RgrSKuLWXMyBH+/l1Dx/I4tXuAJIrlPyo+VmiOc7b5NzHptkSHEPfR9s1
    OK0VqjknclqCJ3Ig86OMEtEXAMPLE==
  </DP>
  <DQ>
    pjPjvSFw+RoaTu0pgCA/jwW/FGyfN6iim1RFbkT4
    z49DZb2IM885f3vf35eLTaEYRYUHQgZtChNEV0TEXAMPLE==
  </DQ>
  <InverseQ>
    nkvOJTg5QtGNgWb9i
   cVtzrL/1pFEOHbJXwEJdU99N+7sMK+1066DL/HSBUCD63qD4USpnf0myc24in0EXAMPLE==</In
verseQ>
  <D>
     Bc7mp7XYHynuPZxChjWNJZIq+A73gm0ASDv6At7F8Vi9r0xUlQe/v0AQS3ycN8QlyR4XMbzMLYk

      3yjxFDXo4ZKQtOGzLGteCU2srANiLv26/imXA8FVidZftTAtLviWQZB
VPTeYIA69ATUYPEq0a5u5wjGy
      UOij9OWyuEXAMPLE=
   </D>
</RSAKeyValue>
```

The following C# code creates a signed URL that uses a canned policy by:

- Creating a policy statement.
- Hashing the policy statement using SHA1, and signing the result using RSA and the private key for your AWS account or for a trusted AWS account that you specify.
- Base64-encoding the hashed and signed policy statement and replacing special characters to make the string safe to use as a URL request parameter.
- Concatenating the applicable values.

For the complete implementation, see the sample at Signature Code in C#.

Example Canned Policy Signing Method in C#

```
public static string ToUrlSafeBase64String(byte[] bytes)
{
    return System.Convert.ToBase64String(bytes)
        .Replace('+', '-')
        .Replace('=', '_')
        .Replace('/', '~');
}

public static string CreateCannedPrivateURL(string urlString,
    string durationUnits, string durationNumber, string pathToPolicyStmnt,
    string pathToPrivateKey, string privateKeyId)
{
    // args[] 0-thisMethod, 1-resourceUrl, 2-seconds-minutes-hours-days
    // to expiration, 3-numberOfPreviousUnits, 4-pathToPolicyStmnt,
    // 5-pathToPrivateKey, 6-PrivateKeyId

    TimeSpan timeSpanInterval = GetDuration(durationUnits, durationNumber);

    // Create the policy statement.
    string strPolicy = CreatePolicyStatement(pathToPolicyStmnt,
        urlString,
        DateTime.Now,
        DateTime.Now.Add(timeSpanInterval),
        "0.0.0.0/0");
    if ("Error!" == strPolicy) return "Invalid time frame." +
        "Start time cannot be greater than end time.";

    // Copy the expiration time defined by policy statement.
    string strExpiration = CopyExpirationTimeFromPolicy(strPolicy);

    // Read the policy into a byte buffer.
    byte[] bufferPolicy = Encoding.ASCII.GetBytes(strPolicy);

    // Initialize the SHA1CryptoServiceProvider object and hash the policy data.

    using (SHA1CryptoServiceProvider
        cryptoSHA1 = new SHA1CryptoServiceProvider())
    {
        bufferPolicy = cryptoSHA1.ComputeHash(bufferPolicy);

        // Initialize the RSACryptoServiceProvider object.
        RSACryptoServiceProvider providerRSA = new RSACryptoServiceProvider();

        XmlDocument xmlPrivateKey = new XmlDocument();

        // Load PrivateKey.xml, which you created by converting your
        // .pem file to the XML format that the .NET framework uses.
        // Several tools are available. We used
        // .NET 2.0 OpenSSL Public and Private Key Parser,
        // http://www.jensign.com/opensslkey/opensslkey.cs.
        xmlPrivateKey.Load(pathToPrivateKey);

        // Format the RSACryptoServiceProvider providerRSA and
        // create the signature.
        providerRSA.FromXmlString(xmlPrivateKey.InnerXml);
        RSAPKCS1SignatureFormatter rsaFormatter =
            new RSAPKCS1SignatureFormatter(providerRSA);
        rsaFormatter.SetHashAlgorithm("SHA1");
        byte[] signedPolicyHash = rsaFormatter.CreateSignature(bufferPolicy);

        // Convert the signed policy to URL-safe Base64 encoding and
        // replace unsafe characters + = / with the safe characters - _ ~
        string strSignedPolicy = ToUrlSafeBase64String(signedPolicyHash);

        // Concatenate the URL, the timestamp, the signature,
        // and the key pair ID to form the signed URL.
        return urlString +
            "?Expires=" +
            strExpiration +
            "&Signature=" +
            strSignedPolicy +
            "&Key-Pair-Id=" +
            privateKeyId;
    }
}
```

The following C# code creates a signed URL that uses a custom policy by:

- Creating a policy statement.
- Base64-encoding the policy statement and replacing special characters to make the string safe to use as a URL request parameter.
- Hashing the policy statement using SHA1, and encrypting the result using RSA and the private key for your AWS account or for a trusted AWS account that you specify.
- Base64-encoding the hashed policy statement and replacing special characters to make the string safe to use as a URL request parameter.
- Concatenating the applicable values.

For the complete implementation, see the sample at Signature Code in C#.

Example Custom Policy Signing Method in C#

```
public static string ToUrlSafeBase64String(byte[] bytes)
{
    return System.Convert.ToBase64String(bytes)
        .Replace('+', '-')
        .Replace('=', '_')
        .Replace('/', '~');
}

public static string CreateCustomPrivateURL(string urlString,
    string durationUnits, string durationNumber, string startIntervalFromNow,

    string ipaddress, string pathToPolicyStmnt, string pathToPrivateKey,
    string PrivateKeyId)
{
    // args[] 0-thisMethod, 1-resourceUrl, 2-seconds-minutes-hours-days
    // to expiration, 3-numberOfPreviousUnits, 4-starttimeFromNow,
    // 5-ip_address, 6-pathToPolicyStmt, 7-pathToPrivateKey, 8-privateKeyId

    TimeSpan timeSpanInterval = GetDuration(durationUnits, durationNumber);
    TimeSpan timeSpanToStart = GetDurationByUnits(durationUnits,
        startIntervalFromNow);
    if (null == timeSpanToStart)
        return "Invalid duration units." +
            "Valid options: seconds, minutes, hours, or days";

    string strPolicy = CreatePolicyStatement(
        pathToPolicyStmnt, urlString, DateTime.Now.Add(timeSpanToStart),
        DateTime.Now.Add(timeSpanInterval), ipaddress);

    // Read the policy into a byte buffer.
    byte[] bufferPolicy = Encoding.ASCII.GetBytes(strPolicy);

    // Convert the policy statement to URL-safe Base64 encoding and
    // replace unsafe characters + = / with the safe characters - _ ~

    string urlSafePolicy = ToUrlSafeBase64String(bufferPolicy);

    // Initialize the SHA1CryptoServiceProvider object and hash the policy data.

    byte[] bufferPolicyHash;
    using (SHA1CryptoServiceProvider cryptoSHA1 =
        new SHA1CryptoServiceProvider())
    {
        bufferPolicyHash = cryptoSHA1.ComputeHash(bufferPolicy);

        // Initialize the RSACryptoServiceProvider object.
        RSACryptoServiceProvider providerRSA = new RSACryptoServiceProvider();

        XmlDocument xmlPrivateKey = new XmlDocument();

        // Load PrivateKey.xml, which you created by converting your
        // .pem file to the XML format that the .NET framework uses.
        // Several tools are available. We used
        // .NET 2.0 OpenSSL Public and Private Key Parser,
        // http://www.jensign.com/opensslkey/opensslkey.cs.
        xmlPrivateKey.Load("PrivateKey.xml");

        // Format the RSACryptoServiceProvider providerRSA
        // and create the signature.
        providerRSA.FromXmlString(xmlPrivateKey.InnerXml);
        RSAPKCS1SignatureFormatter RSAFormatter =
            new RSAPKCS1SignatureFormatter(providerRSA);
        RSAFormatter.SetHashAlgorithm("SHA1");
        byte[] signedHash = RSAFormatter.CreateSignature(bufferPolicyHash);

        // Convert the signed policy to URL-safe Base64 encoding and
        // replace unsafe characters + = / with the safe characters - _ ~
        string strSignedPolicy = ToUrlSafeBase64String(signedHash);

        return urlString +
            "?Policy=" +
            urlSafePolicy +
            "&Signature=" +
            strSignedPolicy +
            "&Key-Pair-Id=" +
            PrivateKeyId;
    }
}
```

Example Utility Methods for Signature Generation

The following methods get the policy statement from a file and parse time intervals for signature generation.

```
public static string CreatePolicyStatement(string policyStmnt,
   string resourceUrl,
   DateTime startTime,
   DateTime endTime,
   string ipAddress)

{
   // Create the policy statement.
   FileStream streamPolicy = new FileStream(policyStmnt, FileMode.Open,
FileAccess.Read);
   using (StreamReader reader = new StreamReader(streamPolicy))
   {
      string strPolicy = reader.ReadToEnd();

      TimeSpan startTimeSpanFromNow = (startTime - DateTime.Now);
      TimeSpan endTimeSpanFromNow = (endTime - DateTime.Now);
      TimeSpan intervalStart =
         (DateTime.UtcNow.Add(startTimeSpanFromNow)) -
         new DateTime(1970, 1, 1, 0, 0, 0, DateTimeKind.Utc);
      TimeSpan intervalEnd =
         (DateTime.UtcNow.Add(endTimeSpanFromNow)) -
         new DateTime(1970, 1, 1, 0, 0, 0, DateTimeKind.Utc);

      int startTimestamp = (int)intervalStart.TotalSeconds; // START_TIME
      int endTimestamp = (int)intervalEnd.TotalSeconds;  // END_TIME

      if (startTimestamp > endTimestamp)
         return "Error!";

      // Replace variables in the policy statement.
      strPolicy = strPolicy.Replace("RESOURCE", resourceUrl);
      strPolicy = strPolicy.Replace("START_TIME", startTimestamp.ToString());
      strPolicy = strPolicy.Replace("END_TIME", endTimestamp.ToString());
      strPolicy = strPolicy.Replace("IP_ADDRESS", ipAddress);
      strPolicy = strPolicy.Replace("EXPIRES", endTimestamp.ToString());
      return strPolicy;
   }
}

public static TimeSpan GetDuration(string units, string numUnits)
{
   TimeSpan timeSpanInterval = new TimeSpan();
   switch (units)
   {
      case "seconds":
         timeSpanInterval = new TimeSpan(0, 0, 0, int.Parse(numUnits));
         break;
      case "minutes":
         timeSpanInterval = new TimeSpan(0, 0, int.Parse(numUnits), 0);
         break;
      case "hours":
         timeSpanInterval = new TimeSpan(0, int.Parse(numUnits), 0 ,0);
         break;
      case "days":
         timeSpanInterval = new TimeSpan(int.Parse(numUnits),0 ,0 ,0);
         break;
      default:
         Console.WriteLine("Invalid time units;" +
            "use seconds, minutes, hours, or days");
         break;
   }
   return timeSpanInterval;
}

private static TimeSpan GetDurationByUnits(string durationUnits,
   string startIntervalFromNow)
{
   switch (durationUnits)
   {
      case "seconds":
         return new TimeSpan(0, 0, int.Parse(startIntervalFromNow));
      case "minutes":
         return new TimeSpan(0, int.Parse(startIntervalFromNow), 0);
      case "hours":
         return new TimeSpan(int.Parse(startIntervalFromNow), 0, 0);
      case "days":
         return new TimeSpan(int.Parse(startIntervalFromNow), 0, 0, 0);
      default:
         return new TimeSpan(0, 0, 0, 0);
   }
}

public static string CopyExpirationTimeFromPolicy(string policyStatement)
{
   int startExpiration = policyStatement.IndexOf("EpochTime");
   string strExpirationRough = policyStatement.Substring(startExpiration +
      "EpochTime".Length);
   char[] digits = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9' };

   List<char> listDigits = new List<char>(digits);
   StringBuilder buildExpiration = new StringBuilder(20);

   foreach (char c in strExpirationRough)
   {
      if (listDigits.Contains(c))
         buildExpiration.Append(c);
   }
   return buildExpiration.ToString();
}
```

See also

- Create a URL Signature Using Perl (p. 153)
- Create a URL Signature Using PHP (p. 154)
- Create a URL Signature Using Java (p. 165)
- Tools for Configuring Private Content (p. 285)

Create a URL Signature Using Java

The Open source Java toolkit for Amazon S3 and CloudFront provides sample code and information about CloudFront development in Java. For information about private distributions, go to Private Distributions at Programmer Guide: Code Samples.

> **Note**
> Creating a URL signature is just one part of the process of serving private content using a signed URL. For more information about the entire process, see How Private Content Works (p. 119).

The following methods are from the Java open source toolkit for Amazon S3 and CloudFront. You must convert the private key from PEM to DER format for Java implementations to use it.

Example Java Policy and Signature Encryption Methods

```
// Signed URLs for a private distribution
// Note that Java only supports SSL certificates in DER format,
// so you will need to convert your PEM-formatted file to DER format.
// To do this, you can use openssl:
// openssl pkcs8 -topk8 -nocrypt -in origin.pem -inform PEM -out new.der
//    -outform DER
// So the encoder works correctly, you should also add the bouncy castle jar
// to your project and then add the provider.

Security.addProvider(new org.bouncycastle.jce.provider.BouncyCastleProvider());

String distributionDomain = "a1b2c3d4e5f6g7.cloudfront.net";
String privateKeyFilePath = "/path/to/rsa-private-key.der";
String s3ObjectKey = "s3/object/key.txt";
String policyResourcePath = "http://" + distributionDomain + "/" + s3ObjectKey;

// Convert your DER file into a byte array.

byte[] derPrivateKey = ServiceUtils.readInputStreamToBytes(new
    FileInputStream(privateKeyFilePath));

// Generate a "canned" signed URL to allow access to a
// specific distribution and object

String signedUrlCanned = CloudFrontService.signUrlCanned(
    "http://" + distributionDomain + "/" + s3ObjectKey, // Resource URL or Path

    keyPairId,     // Certificate identifier,
                   // an active trusted signer for the distribution
    derPrivateKey, // DER Private key data
    ServiceUtils.parseIso8601Date("2011-11-14T22:20:00.000Z") // DateLessThan
    );
System.out.println(signedUrlCanned);

// Build a policy document to define custom restrictions for a signed URL.

String policy = CloudFrontService.buildPolicyForSignedUrl(
    // Resource path (optional, may include '*' and '?' wildcards)
    policyResourcePath,
    // DateLessThan
    ServiceUtils.parseIso8601Date("2011-11-14T22:20:00.000Z"),
    // CIDR IP address restriction (optional, 0.0.0.0/0 means everyone)
    "0.0.0.0/0",
    // DateGreaterThan (optional)
    ServiceUtils.parseIso8601Date("2011-10-16T06:31:56.000Z")
    );

// Generate a signed URL using a custom policy document.

String signedUrl = CloudFrontService.signUrl(
    // Resource URL or Path
    "http://" + distributionDomain + "/" + s3ObjectKey,
    // Certificate identifier, an active trusted signer for the distribution
    keyPairId,
    // DER Private key data
    derPrivateKey,
    // Access control policy
    policy
    );
System.out.println(signedUrl);
```

See also

- Create a URL Signature Using Perl (p. 153)
- Create a URL Signature Using PHP (p. 154)
- Create a URL Signature Using C# and the .NET Framework (p. 157)
- Tools for Configuring Private Content (p. 285)

Using an HTTPS Connection to Access Your Objects

Topics

- How CloudFront Works with HTTPS Connections (p. 169)
- How to Require HTTPS for Communication between Viewers, CloudFront, and Your Origin (p. 169)
- Using Alternate Domain Names and HTTPS (p. 171)
- Charges for HTTPS Connections (p. 177)

For web distributions, you can use HTTPS requests to ensure that your objects are encrypted when CloudFront serves them to viewers and, optionally, when CloudFront gets the objects from your origin:

- **To require HTTPS between CloudFront and viewers:** Configure some or all of your CloudFront cache behaviors either to redirect HTTP requests to HTTPS requests or to require that viewers use only the HTTPS protocol to access your objects in the CloudFront cache. You can also configure one or more cache behaviors in the same distribution to allow both HTTP and HTTPS, so you can require HTTPS for some objects but not for others.
- **To require HTTPS between CloudFront and your origin (optional):** Configure one or more CloudFront origins to require that CloudFront fetches objects from your origin using the protocol that the viewer used to request the objects. For example, when you use this CloudFront setting and the viewer uses HTTPS to request an object from CloudFront, CloudFront also uses HTTPS to forward the request to your origin. When your origin is an Amazon S3 bucket, this is the default setting and cannot be changed.

  Important

  If your Amazon S3 bucket is configured as a website endpoint, you cannot configure CloudFront to use HTTPS to communicate with your origin because Amazon S3 doesn't support HTTPS connections in that configuration.

If you're using an HTTP server as your origin, and if you want to use HTTPS both between viewers and CloudFront and between CloudFront and your origin, you must install an SSL certificate on the HTTP server that is signed by a third-party certificate authority, for example, VeriSign or DigiCert.

Caution

If the origin server returns an invalid certificate or a self-signed certificate, or if the origin server returns the certificate chain in the wrong order, CloudFront drops the TCP connection, returns HTTP error code 502, and sets the `X-Cache` header to `Error from cloudfront`.

How CloudFront Works with HTTPS Connections

The following example of how CloudFront works with HTTPS connections assumes the following:

- Your CloudFront distribution has one cache behavior (the default cache behavior) and one origin.
- You have configured your distribution to use HTTPS between viewers and CloudFront and between CloudFront and your origin.
- Your origin has an SSL certificate that was signed by a third-party certificate authority.

The process works basically the same way whether your origin server is an Amazon S3 bucket or an HTTP server.

CloudFront Process for Serving Objects Using HTTPS

1. A viewer submits an HTTPS request to CloudFront. There's some SSL negotiation here between the viewer and CloudFront. In the end, the viewer submits the request in an encrypted format.
2. If the object is in the CloudFront edge cache, CloudFront encrypts the object and returns it to the viewer, and the viewer decrypts it.
3. If the object is not in the CloudFront cache, CloudFront performs the SSL negotiation with your origin and, when the negotiation is complete, forwards the request to your origin in an encrypted format.
4. Your origin decrypts the request, encrypts the requested object, and returns the object to CloudFront.
5. CloudFront decrypts the object, re-encrypts it, and forwards the object to the viewer. CloudFront also saves the object in the edge cache so the object is available next time it's requested.
6. The viewer decrypts the object.

How to Require HTTPS for Communication between Viewers, CloudFront, and Your Origin

You can configure CloudFront to require HTTPS for communication between viewers and CloudFront and, optionally, between CloudFront and your origin.

> **Note**
> To ensure that objects are encrypted from the origin to CloudFront edge caches and from edge caches to viewers, use only HTTPS. If you ever configure CloudFront to get objects from your origin using HTTP, CloudFront adds the objects to the edge cache and continues to serve them to viewers until the objects expire, or until you remove or replace them. For more information about removing or replacing objects in a distribution, see Adding, Removing, or Replacing Objects in a Distribution (p. 81).

If you want to use alternate domain names (for example, `example.com`) instead of the domain name that CloudFront assigns to your distribution, also see Using Alternate Domain Names and HTTPS (p. 171).

To Require HTTPS for Communication between Viewers, CloudFront, and Your Origin

1. (Optional) If you want CloudFront to use HTTPS when communicating with a custom origin, get and install an SSL certificate from a third-party certificate authority such as VeriSign or DigiCert (if you don't already have one).

   > **Important**
   > If you configure CloudFront to use HTTPS when communicating with your origin in step 3, CloudFront verifies that your certificate was issued by an established third-party certificate

authority. CloudFront supports the same certificate authorities as Mozilla; for the current list, see Mozilla Included CA Certificate List. You cannot use a self-signed certificate.

If you're using an Amazon S3 bucket, Amazon S3 provides an SSL certificate.

For more information about getting and installing an SSL certificate, refer to the documentation for your HTTP server software and to the documentation for the third-party certificate authority.

2. Configure your distribution either to redirect HTTP requests to HTTPS or to require that viewers use HTTPS when communicating with CloudFront. To do this in the CloudFront console, create or update one or more cache behaviors in your distribution to have one of the following settings for **Viewer Protocol Policy**:

   - **Redirect to HTTPS**: If a user sends an HTTP request instead of an HTTPS request, CloudFront returns an HTTP status code of 301 (Moved Permanently) along with the new HTTPS URL. The viewer then resubmits the request to CloudFront using the HTTPS URL.

     Note

     When a viewer makes an HTTP request that is redirected to an HTTPS request, CloudFront charges for both requests. For the HTTP request, the charge is only for the request and for the headers that CloudFront returns to the viewer. For the HTTPS request, the charge is for the request, and for the headers and the object returned by your origin.

   - **HTTPS Only**: If a user sends an HTTP request instead of an HTTPS request, CloudFront returns an HTTP status code of 403 (Forbidden) and does not return the object.

   For information about using the CloudFront console to update a web distribution, see Listing, Viewing, and Updating CloudFront Distributions (p. 27).

   For information about using the CloudFront API to update a web distribution, go to PUT Distribution Config in the *Amazon CloudFront API Reference*. If you're using the API, see the `ViewerProtocol-Policy` element.

3. (Optional) To require that CloudFront uses HTTPS when communicating with your origin, create or update one or more origins in your distribution to have the following settings:

   - **CloudFront Console:** For **Origin Protocol Policy**, specify **Match Viewer**.
   - **CloudFront API:** For `OriginProtocolPolicy`, specify `match-viewer`.

   When your origin is an Amazon S3 bucket, **Match Viewer** is the default setting and cannot be changed.

   Important

   When you're using a custom origin, the SSL certificate on your origin includes a domain name in the Common Name field and possibly several more in the Subject Alternative Names field. (CloudFront supports wildcard characters in certificate domain names.) One of the domain names in the certificate must match the domain name that you specify for Origin Domain Name. If the domain names don't match, the following sequence occurs when an end user submits an HTTPS request for an object:

   a. CloudFront forwards the request to the applicable origin based on the settings in your cache behaviors.
   b. The origin returns an SSL certificate.
   c. CloudFront inspects the certificate and determines that the domain names in the certificate don't match the value of Origin Domain Name in your distribution.
   d. CloudFront returns an HTTP status code 502 (bad gateway) to the end user.

4. Confirm the following:

- The path pattern in each cache behavior applies only to the requests for which you want viewers to use HTTPS.
- The cache behaviors are listed in the desired order. For more information, see Path Pattern (p. 45).
- The cache behaviors are routing requests to the origins for which you have configured an **Origin Protocol Policy** of **Match Viewer**, if applicable.
- If you're using a custom origin and you configured CloudFront to use HTTPS when communicating with the origin, the origin must have a valid certificate signed by a third-party certificate authority.

5. Test the configuration before you use it in a production environment.

Using Alternate Domain Names and HTTPS

Topics

- Choosing How CloudFront Serves HTTPS Requests (p. 171)
- Requirements and Limits on Using SSL Certificates with CloudFront (p. 172)
- To use alternate domain names with HTTPS (p. 174)
- Determining the Size of the Public Key in an SSL Certificate (p. 175)
- Rotating SSL Certificates (p. 175)
- Reverting from a Custom SSL Certificate to the Default CloudFront Certificate (p. 176)
- Switching from a Custom SSL Certificate with Dedicated IP Addresses to SNI (p. 177)

By default, you can deliver your content to viewers over HTTPS by using your CloudFront distribution domain name in your URLs, for example, `https://d111111abcdef8.cloudfront.net/image.jpg`. For more information, see How to Require HTTPS for Communication between Viewers, CloudFront, and Your Origin (p. 169).

If you want your viewers to use HTTPS and you want to use your own domain name in the URLs for your objects (for example, `https://www.example.com/image.jpg`), you need to perform several additional steps, as explained in this topic.

> **Important**
> When you add a certificate to your distribution, CloudFront immediately propagates the certificate to all of its edge locations. As new edge locations become available, CloudFront will propagate the certificate to those locations, too. You cannot restrict the edge locations to which CloudFront propagates your certificates.

Choosing How CloudFront Serves HTTPS Requests

If you want your users to use HTTPS and to use alternate domain names for your objects, you need to choose how CloudFront serves HTTPS requests. When you configure CloudFront to use alternate domain names, CloudFront can serve HTTPS requests either by using a dedicated IP address in each edge location or by using Server Name Indication (SNI).

Serving HTTPS Requests Using Dedicated IP Addresses (Works for All Clients)

If you configure CloudFront to serve HTTPS requests using dedicated IP addresses, CloudFront associates your alternate domain name with a dedicated IP address in each CloudFront edge location. When a viewer submits an HTTPS request for your content, DNS routes the request to the IP address for your distribution in the applicable edge location. CloudFront uses the IP address to identify your distribution and to determine which SSL certificate to return to the viewer. The viewer and CloudFront perform SSL negotiation using your SSL certificate, and CloudFront returns the requested content to the viewer. This method works for every HTTPS request, regardless of the browser or other viewer that the user is using.

> **Important**
> If you configure CloudFront to serve HTTPS requests using dedicated IP addresses, you incur an additional monthly charge. The charge begins when you associate your SSL certificate with a distribution and you enable the distribution. For more information about CloudFront pricing, see Amazon CloudFront Pricing.

Serving HTTPS Requests Using SNI (Works for Most Clients)

If you configure CloudFront to serve HTTPS requests using Server Name Indication (SNI), CloudFront associates your alternate domain name with an IP address for each edge location, but the IP address is not dedicated to your distribution. When a viewer submits an HTTPS request for your content, DNS routes the request to the IP address for the applicable edge location. However, because the IP address isn't dedicated to your distribution, CloudFront can't determine, based on the IP address, which domain the request is for.

SSL negotiation occurs very early in the process of establishing an HTTPS connection. If CloudFront can't immediately determine which domain the request is for, it drops the connection. Using a dedicated IP address is one way to associate a request with a domain. The other is Server Name Indication (SNI), which is an extension to the TLS protocol that is supported by most modern browsers. Browsers that support SNI automatically get the domain name from the request URL and add it to a new field in the request header. When CloudFront receives an HTTPS request from a browser that supports SNI, it finds the domain name in the request header and responds to the request with the applicable SSL certificate. The viewer and CloudFront perform SSL negotiation, and CloudFront returns the requested content to the viewer.

For a current list of the browsers that support SNI, see the Wikipedia entry Server Name Indication.

If you want to use SNI but some of your users' browsers don't support SNI, you have several options:

- Configure CloudFront to serve HTTPS requests by using dedicated IP addresses instead of SNI.
- Use the CloudFront SSL certificate instead of a custom certificate. This requires that you use the CloudFront domain name for your distribution in the URLs for your objects, for example, `https://d111111abcdef8.cloudfront.net/logo.png`.

  You also need to change the SSL certificate that CloudFront is using from a custom certificate to the default CloudFront certificate:
  - If you haven't used your distribution to distribute your content, you can just change the configuration. For more information, see Listing, Viewing, and Updating CloudFront Distributions (p. 27).
  - If you have used your distribution to distribute your content, you need to create a new CloudFront distribution and change the URLs for your objects to reduce or eliminate the amount of time that your content is unavailable. For more information, see Reverting from a Custom SSL Certificate to the Default CloudFront Certificate (p. 176).
- If you can control which browser your users use, have them upgrade their browser to one that supports SNI.
- Use HTTP instead of HTTPS.

Requirements and Limits on Using SSL Certificates with CloudFront

Note the following requirements for your certificate:

- Your certificate must be issued by a recognized certificate authority (CA). Self-signed certificates are not accepted.
- Your certificate must be in X.509 PEM format.
- In the .pem file, list all of the intermediate certificates in the certificate chain, beginning with one for the CA that signed the certificate for your domain. Typically, you'll find a file on your CA's website that lists intermediate and root certificates in the proper chained order.

  Important

  Do not include the root certificate, intermediate certificates that are not in the trust path, or your CA's public key certificate.

  Here's an example:

  ```
  -----BEGIN CERTIFICATE-----
  Intermediate certificate 2
  -----END CERTIFICATE-----
  -----BEGIN CERTIFICATE-----
  Intermediate certificate 1
  -----END CERTIFICATE-----
  ```

- The private key must match the public key that is in the certificate. It must also be an RSA private key in PEM format, where the PEM header is `BEGIN RSA PRIVATE KEY` and the footer is `END RSA PRIVATE KEY`. The private key cannot be encrypted with a password.
- You must have permission to use and upload the SSL certificate, including permission from the certificate authority that issued the certificate to upload it onto a content delivery network.
- The maximum size of the public key in an SSL certificate is 2048 bits. For information about how to determine the size of the public key, see Determining the Size of the Public Key in an SSL Certificate (p. 175).
- CloudFront supports all types of certificates including domain-validated certificates, extended validation (EV) certificates, high-assurance certificates, wildcard certificates (`*.example.com`), subject alternative name (SAN) certificates (`example.com` and `example.net`), and so on.
- You are responsible for monitoring certificate expiration dates and for renewing SSL certificates that you upload and use with CloudFront.

In addition, note the following limits on using SSL certificates with CloudFront:

- You can associate a maximum of one SSL certificate with each CloudFront distribution.
- You can upload a maximum of 10 SSL certificates to the IAM certificate store for each AWS account. To request a higher limit, go to Request IAM limit increase.
- If you want to use the same certificate with multiple CloudFront distributions that were created with different accounts, you must upload the certificate to the IAM certificate store once for each AWS account.
- If you want to use the same certificate both for CloudFront and for other AWS services, you must upload the certificate twice: once for CloudFront and once for the other services. For information about how to upload the certificate for CloudFront, see the following procedure.
- When you're using a custom origin, the SSL certificate on your origin includes a domain name in the Common Name field and possibly several more in the Subject Alternative Names field. (CloudFront supports wildcard characters in certificate domain names.) One of the domain names in the certificate must match the domain name that you specify for Origin Domain Name. If the domain names don't match, CloudFront returns an HTTP status code 502 (bad gateway) to the end user.
- If you want to serve HTTPS requests using dedicated IP addresses, note the following:
  - You must request permission to use dedicated IP addresses. See the following procedure.
  - Unless you specify otherwise, AWS gives you permission to use two certificates with your AWS account, one for everyday use and one for when you need to rotate certificates for multiple distributions.

- If you already have been approved to use this feature but need to increase the number of custom SSL certificates that you can use with your AWS account, go to https://aws.amazon.com/support/createCase?type=service_limit_increase&serviceLimitIncreaseType=cloudfront-distributions.

To use alternate domain names with HTTPS

1. If you want to serve HTTPS requests using SNI, skip to step 2.

   If you want to serve HTTPs requests using dedicated IP addresses, request permission for your AWS account. We'll update your account as soon as possible. For more information and to request permission, see Custom SSL Certificates for Amazon CloudFront.

   Important
   By default, when you request permission to use an alternate domain name with HTTPS, AWS updates your account so you can associate two custom SSL certificates with your CloudFront distributions. Typically, you'll use the second certificate only temporarily, when you have more than one distribution and you need to rotate certificates. If you need to permanently associate two or more certificates with your distributions, indicate how many certificates you need and describe the circumstances in your request.

2. Use the AWS CLI to upload your SSL certificate to the IAM certificate store. If you don't already have your certificate, see Creating, Uploading, and Deleting Server Certificates in *Using IAM*.

   If you already have your certificate, use the following AWS CLI command to upload a signed certificate:

   ```
   aws iam upload-server-certificate --server-certificate-name CertificateName
    --certificate-body file://public_key_certificate_file --private-key
   file://privatekey.pem --certificate-chain file://certificate_chain_file -
   -path /cloudfront/path/
   ```

   Note the following:

   - You must upload your certificate to the IAM certificate store using the same AWS account that you used to create your CloudFront distribution.
   - When you upload your certificate to IAM, the value of the `-path` parameter (certificate path) must start with `/cloudfront/`, for example, `/cloudfront/production/` or `/cloudfront/test/`. The path also must end with a /.
   - If you plan to use the CloudFront console to create or update your distribution, the value that you specify for the `--server-certificate-name` parameter in the AWS CLI is the value that will appear in the **SSL Certificate** list in the CloudFront console.
   - If you plan to use the CloudFront API to create or update your distribution, make note of the alphanumeric string that the AWS CLI returns, for example `AS1A2M3P4L5E67SIIXR3J`. This is the value that you will specify in the `IAMCertificateId` element. You don't need the IAM ARN, which is also returned by the CLI.

   For more information about the AWS CLI, go to the *AWS Command Line Interface User Guide* and the *AWS Command Line Interface Reference*.

3. Update your distribution to include your alternate domain names, to specify which SSL certificate you want to use, and to specify whether you want CloudFront to use dedicated IP addresses or SNI to serve HTTPS requests. You also need to add or update DNS records. For more information and a procedure, see Using Alternate Domain Names (CNAMEs) (p. 29).

   Caution
   After you associate your SSL certificate with your CloudFront distribution, do not delete the certificate from the IAM certificate store until you remove the certificate from all distributions and until the status of the distributions has changed to **Deployed**.

Determining the Size of the Public Key in an SSL Certificate

When you're using CloudFront alternate domain names and HTTPS, the size of the public key in an SSL certificate cannot exceed 2048 bits. (This is not the number of characters in the public key.) You can determine the size of the public key by running the following OpenSSL command:

```
openssl x509 -in path and filename of SSL certificate -text -noout
```

where:

- `-in` specifies the path and filename of your SSL certificate.
- `-text` causes OpenSSL to display the length of the public key in bits.
- `-noout` prevents OpenSSL from displaying the public key.

Example output:

```
Public-Key: (2048 bit)
```

Rotating SSL Certificates

Occasionally you'll need to replace one SSL certificate with another, because, for example, the expiration date is approaching. The process depends on whether you have associated your SSL certificate with one or more CloudFront distributions under the same AWS account:

- **SSL certificate associated with one distribution:** You can just update your distribution and replace the old certificate with the new one. For more information, see Listing, Viewing, and Updating CloudFront Distributions (p. 27).
- **SSL certificate associated with two or more distributions under the same AWS account:** By default, when you request permission to use an alternate domain name with HTTPS, you can associate only two SSL certificates with the CloudFront distributions under one AWS account. Typically, you'll use the second certificate only when you have more than one distribution and you need to rotate certificates. One certificate is associated with distributions that you haven't updated yet, and the other certificate is associated with distributions that you have updated. Perform the following procedure.

  Important
  While you are rotating certificates, you might incur an additional, pro-rated charge for using the second certificate. We recommend that you update your distributions promptly to minimize the additional charge.

To rotate SSL certificates for two or more CloudFront distributions

1. If you configured CloudFront to use dedicated IP addresses to serve HTTPS requests and you have already associated the maximum number of SSL certificates permitted by AWS for your account, request permission to associate an additional certificate. Send an email to cloudfront-ssl-request@amazon.com, and explain that you are rotating certificates.
2. Update your distributions one at a time to use a new certificate.

   If you submitted a request to AWS in Step 1, wait until you receive notification that your AWS account has been updated.

   For more information, see Listing, Viewing, and Updating CloudFront Distributions (p. 27).

3. (Optional) After you have updated all of your CloudFront distributions, you can delete the old certificate from the IAM certificate store.

   Caution
   Do not delete an SSL certificate from the IAM certificate store until you remove it from all distributions and until the status of the distributions that you have updated has changed to **Deployed**.

Reverting from a Custom SSL Certificate to the Default CloudFront Certificate

If you configured CloudFront to use a custom SSL certificate and you want to change your configuration to use CloudFront's SSL certificate, the process depends on whether you've used your distribution to distribute your content:

- If you have not used your distribution to distribute your content, you can just change the configuration. For more information, see Listing, Viewing, and Updating CloudFront Distributions (p. 27).
- If you have used your distribution to distribute your content, you need to create a new CloudFront distribution and change the URLs for your objects to reduce or eliminate the amount of time that your content is unavailable. Perform the following procedure.

To revert to the default CloudFront certificate

1. Create a new CloudFront distribution with the desired configuration. For **SSL Certificate**, choose **Default CloudFront Certificate (*.cloudfront.net)**.

   For more information, see Task List for Creating a Web Distribution (p. 36).
2. For objects that you're distributing using CloudFront, update the URLs in your application to use the domain name that CloudFront assigned to the new distribution. For example, change `https://www.example.com/images/logo.png` to `https://d111111abcdef8.cloudfront.net/images/logo.png`.
3. Either delete the distribution that is associated with a custom SSL certificate, or update the distribution to change the value of **SSL Certificate** to **Default CloudFront Certificate (*.cloudfront.net)**. For more information, see Listing, Viewing, and Updating CloudFront Distributions (p. 27).

   Important
   Until you complete this step, Amazon Web Services continues to charge you for using a custom SSL certificate.
4. (Optional) Use the AWS CLI to delete your custom SSL certificate from the IAM certificate store. This is the same application that you used to add the custom SSL certificate to the IAM certificate store:

   a. Run the AWS CLI command `list-signing-certificates` to get the certificate ID of the certificate that you want to delete. For more information, see list-signing-certificates in the *AWS Command Line Interface Reference*.
   b. Run the AWS CLI command `delete-signing-certificate`to delete the certificate. For more information, see delete-signing-certificate in the *AWS Command Line Interface Reference*.

Switching from a Custom SSL Certificate with Dedicated IP Addresses to SNI

If you configured CloudFront to use a custom SSL certificate with dedicated IP addresses, you can switch to using a custom SSL certificate with SNI instead. The following procedure shows you how.

> **Important**
> This update to your CloudFront configuration has no effect on viewers that support SNI; they can access your content before and after the change, as well as while the change is propagating to CloudFront edge locations. Viewers that don't support SNI cannot access your content after the change. For more information, see Choosing How CloudFront Serves HTTPS Requests (p. 171).

To switch from a custom SSL certificate with dedicated IP addresses to SNI

1. Sign in to the AWS Management Console and open the Amazon CloudFront console at https://console.aws.amazon.com/cloudfront/.
2. In the top pane of the CloudFront console, select the distribution that you want to view or update.
3. Click **Distribution Settings**.
4. On the **General** tab, click **Edit**.
5. Change the setting of **Custom SSL Client Support** to **Only Clients that Support Server Name Indication (SNI)**.
6. Click **Yes, Edit**.

Charges for HTTPS Connections

You always incur a surcharge for HTTPS requests. For more information, see Amazon CloudFront Pricing.

Using IAM to Control Access to CloudFront Resources

Topics

- CloudFront Resources (p. 178)
- CloudFront Actions (p. 179)
- Policy Keys (p. 180)
- Example Policies for CloudFront (p. 180)

Amazon CloudFront integrates with AWS Identity and Access Management (IAM) so that you can create Users for your AWS Account and you can specify which CloudFront actions a User (or a group of Users) can perform in your AWS Account. You control User access to CloudFront by creating policies that describe User or group permissions. For example, you might create a policy that gives only certain Users in your organization permission to use `GetDistributionConfig`. They could then use the action to retrieve data about your CloudFront distributions.

For more information on using policies to set AWS Account User permissions, go to Permissions and Policies in *Using AWS Identity and Access Management*. For general information about IAM, go to AWS Identity and Access Management on the AWS website.

> **Important**
> Using Amazon CloudFront with IAM doesn't change how you use CloudFront. There are no changes to CloudFront actions, and no new CloudFront actions related to Users and access control.

CloudFront Resources

You use an asterisk (*) as the resource when writing a policy to control access to CloudFront actions. This is because you can't use IAM to control access to specific CloudFront resources. For example, you can't give Users access to a specific distribution. Permissions granted using IAM include all the resources you use with CloudFront. Because you cannot specify the resources to control access to, there are no CloudFront resource ARNs (Amazon Resource Names) for you to use in an IAM policy. (For detailed information about using ARNs with IAM, go to "ARNs" in the Identifiers for IAM Entities section of *Using AWS Identity and Access Management*.)

CloudFront Actions

In an IAM policy, you can specify any and all API actions that CloudFront offers. The action name must be prefixed with the lowercase string `cloudfront:`. For example: `cloudfront:GetDistribution-Config`, `cloudfront:ListInvalidations`, or `cloudfront:*` (for all CloudFront actions).

The following tables list the canonical names for all CloudFront actions. Use these canonical names when specifying APIs in IAM policies.

Web Distributions

API Actions for Web Distributions	Canonical Name
POST Distribution	CreateDistribution
GET Distribution	GetDistribution
GET Distribution Config	GetDistributionConfig
PUT Distribution Config	UpdateDistribution
GET Distribution List	ListDistributions
DELETE Distribution	DeleteDistribution

RTMP Distributions

API Actions for RTMP Distributions	Canonical Name
POST Streaming Distribution	CreateStreamingDistribution
GET Streaming Distribution	GetStreamingDistribution
GET Streaming Distribution Config	GetStreamingDistributionConfig
PUT Streaming Distribution Config	UpdateStreamingDistribution
GET Streaming Distribution List	ListStreamingDistributions
DELETE Streaming Distribution	DeleteStreamingDistribution

Invalidations

API Actions for Invalidations	Canonical Name
POST Invalidation	CreateInvalidation
GET Invalidation	GetInvalidation
GET Invalidation List	ListInvalidations

Origin Access Identities

API Action for Origin Access Identities	Canonical Name
POST Origin Access Identity	CreateCloudFrontOriginAccessIdentity
GET Origin Access Identity	GetCloudFrontOriginAccessIdentity
GET Origin Access Identity Config	GetCloudFrontOriginAccessIdentityConfig
PUT Origin Access Identity Config	UpdateCloudFrontOriginAccessIdentity
GET Origin Access Identity List	ListCloudFrontOriginAccessIdentities
DELETE Origin Access Identity	DeleteCloudFrontOriginAccessIdentity

Policy Keys

Policy keys enable you to add conditions to your policies, such as request date or IP range. CloudFront implements the AWS-wide policy keys, but no others. For more information about policy keys, see "Condition" in the Element Descriptions section of *Using AWS Identity and Access Management*.

Example Policies for CloudFront

This section shows a few simple policies for controlling user access to CloudFront.

> **Note**
> In the future, CloudFront might add new actions that should logically be included in one of the following policies, based on the policy's stated goals.

Example 1: Allow a group read and write access to all of resources owned by the account

This example creates a policy that is attached to a group (for example, the Developers group) to give the group read and write access to all CloudFront resources.

```
{
   "Version": "2012-10-17",
   "Statement":[{
      "Effect":"Allow",
      "Action":["cloudfront:*"],
      "Resource":"*"
      }
   ]
}
```

Example 2: Allow a group read access to all of resources owned by the account

This example creates a policy that is attached to a group (for example, the Finance group) to give the group read access to all CloudFront resources.

```
{
   "Version": "2012-10-17",
   "Statement":[{
      "Effect":"Allow",
      "Action":["cloudfront:Get*", "cloudfront:List*"],
      "Resource":"*"
      }
   ]
}
```

Example 3: Allow a group read and write access to all distributions owned by the account

This example creates a policy that is attached to a group (for example, the Ops group) to give the group read and write access to all distributions, but not access to invalidations or origin access identities.

```
{
   "Version": "2012-10-17",
   "Statement":[{
      "Effect":"Allow",
      "Action":["cloudfront:*Distribution*"],
      "Resource":"*"
      }
   ]
}
```

Example 4: Allow a group to retrieve CloudFront distribution data, but only if they're using SSL with the request

This example creates a policy that is attached to a group to give the group access to all CloudFront actions, with a condition that requires use of SSL.

```
{
   "Version": "2012-10-17",
   "Statement":[{
      "Effect":"Allow",
      "Action":["cloudfront:*"],
      "Resource":"*",
      "Condition":{
         "Bool":{
            "aws:SecureTransport":"true"
            }
         }
      }
   ]
}
```

Access Logs

Topics

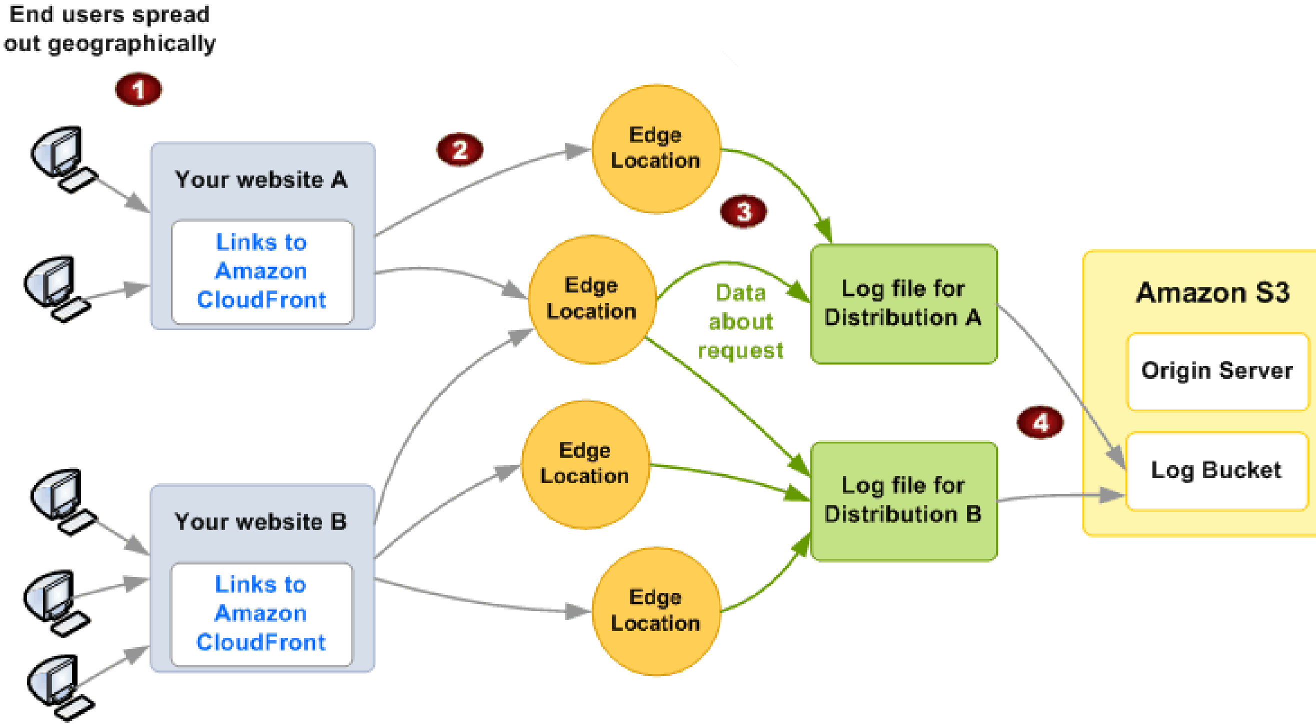
- Overview (p. 182)
- Analyzing Access Logs (p. 184)
- Bucket and File Ownership (p. 184)
- How to Change Logging Settings (p. 184)
- How to Delete Log Files from an Amazon S3 Bucket (p. 185)
- File Name Format and Timing of File Delivery (p. 185)
- Log File Format (p. 185)
- Charges for Access Logs (p. 191)

Amazon CloudFront provides optional log files with information about user requests. This section describes how to enable and disable logging, the content of log files, and how AWS charges you if you decide to use logging.

> **Note**
> If you use a custom origin, you will need to create an Amazon S3 bucket to store your log files in.

Overview

You can enable CloudFront to deliver access logs per distribution to an Amazon S3 bucket of your choice. The following figure and table describe the basic process for access logs.


End users spread out geographically
1
2
3
4
Your website A
Links to Amazon CloudFront
Your website B
Links to Amazon CloudFront
Edge Location
Edge Location
Edge Location
Edge Location
Data about request
Log file for Distribution A
Log file for Distribution B
Amazon S3
Origin Server
Log Bucket

Process for Access Logs

1	Your end users use your application or website. In this graphic, you have two different websites, A and B, each using a different CloudFront distribution (Distribution A and Distribution B).
2	Your end users send requests, and CloudFront routes each request to the appropriate edge location.
3	CloudFront writes data about each request to a log file specific to that distribution. In this example, Information about requests related to Distribution A goes into a log file just for Distribution A, and information about requests related to Distribution B goes into a log file just for Distribution B.
4	CloudFront periodically puts the distribution's log file in an Amazon S3 bucket of your choice, and then starts writing a new log file for the distribution.

Each entry in a log file gives details about a single request. For more information about log file format, see Log File Format (p. 185).

You can store log files for a distribution either in the same Amazon S3 bucket as your origin server or in a different bucket. (To simplify maintenance, we recommend that you use a separate bucket.) You can also store the log files for multiple distributions in the same bucket. When you enable logging for a particular distribution, you can specify an optional prefix for the names of your log files.

If no users access your content during a given hour, you don't receive any log files for that hour.

Analyzing Access Logs

Because logs for a single stream can get recorded in multiple files, we recommend that you combine all the log files you receive for a given period into one file. You can then analyze the data for that period more quickly and accurately.

Important
We recommend that you use the logs to understand the nature of the requests for your content, not as a complete accounting of all requests. CloudFront delivers access logs on a best-effort basis. The log record for a particular request might be delivered long after the request was actually processed, or not at all. In rare cases, usage that appears in the AWS usage tracking and billing systems might not appear in CloudFront access logs.

For more information about CloudFront access logs, including recommendations for tools that you can use to analyze access logs, see Using CloudFront Logging (p. 284).

Bucket and File Ownership

You must have Amazon S3 `FULL_CONTROL` permission for the log file bucket. You have this permission by default if you're the bucket owner. If you're not, the bucket owner must grant your AWS account `FULL_CONTROL` permission.

When you enable logging, you do it with an API call to the CloudFront API. Making that API call also automatically calls the Amazon S3 API to update the bucket's ACL to allow read and write permissions for the *AWS data feeds account*. This account writes the log files to the bucket.

Each log file has its own ACL (separate from the bucket's ACL). The bucket owner has `FULL_CONTROL` permission for the log files, the distribution owner (if not the bucket owner) has no permission, and the data feeds account has read and write permission.

Note
Removing the permissions for the data feeds account does not disable logging. If you remove those permissions, but don't disable logging (which you do with the API), we reinstate those permissions the next time the data feeds account needs to write a log file to your log bucket.

If you disable logging, we don't remove the read/write permissions for the data feeds account on either the bucket or the log files. If you want, you can do that yourself.

How to Change Logging Settings

You can enable or disable logging, change the Amazon S3 bucket where your logs are stored, and change the prefix for log files using the CloudFront console or using the CloudFront API:

- For information about updating a web or an RTMP distribution using the CloudFront console, see Listing, Viewing, and Updating CloudFront Distributions (p. 27).
- For information about updating a web distribution using the CloudFront API, go to PUT Distribution Config in the *Amazon CloudFront API Reference*.
- For information about updating an RTMP distribution using the CloudFront API, go to PUT Streaming Distribution Config in the *Amazon CloudFront API Reference*.

Your changes to logging settings take effect within 12 hours.

To use the CloudFront API to change access log settings for:

- web distributions, you must use the 2009-04-02 or later version of the API.
- RTMP distributions, you must use the 2010-05-01 or later version of the API.
- cookies, you must use the 2012-07-01 or later version of the API.

> **Note**
> To enable easier listing of keys in a bucket, Amazon S3 users commonly use a prefix followed by a slash (/) as a delimiter.

How to Delete Log Files from an Amazon S3 Bucket

CloudFront does not automatically delete log files from the Amazon S3 bucket that you specified when you enabled logging. For information about deleting log files from an Amazon S3 bucket, see the applicable Amazon S3 documentation:

- Using the Amazon S3 console: See Deleting an Object in the *Amazon Simple Storage Service Console User Guide*.
- Using the REST API: See DELETE Object in the *Amazon Simple Storage Service API Reference*.
- Using the SOAP API: See DeleteObject in the *Amazon Simple Storage Service API Reference*.

File Name Format and Timing of File Delivery

The filename follows this format (with the date and hour in UTC):

```
{bucket-name}.s3.amazonaws.com/{optional-prefix/}{distribution-ID}.{YYYY}-{MM}-
{DD}-{HH}.{unique-ID}.gz
```

For example, if your bucket name is `mylogs` and your prefix is `myprefix/`, your filenames look similar to this:

```
mylogs.s3.amazonaws.com/myprefix/EMLARXS9EXAMPLE.2012-07-01-20.RT4KCN4SGK9.gz
```

If you include a value for `{optional-prefix/}`, and if your value doesn't include a `/`, CloudFront adds one automatically. If your value does include a `/`, CloudFront does not add another one.

CloudFront compresses each log file using gzip and saves it in the Amazon S3 bucket that you specify. Typically, CloudFront saves log files within 24 hours after receiving the corresponding requests. Depending on how many users submit requests, CloudFront can save several log files per hour.

> **Note**
> If no users submit requests during a given hour, you don't receive any log files for that hour.

Log File Format

Topics

- Web Distribution Log File Format (p. 186)
- RTMP Distribution Log File Format (p. 189)

Each entry in a log file gives details about a single end user request. The log files for web and for RTMP distributions are not identical, but both types of log files:

- Use the W3C extended log file format. (For more information, go to http://www.w3.org/TR/WD-log-file.html.)
- Contain tab-separated values.
- Contain records that are not necessarily in chronological order.
- Contain two header lines: one with the file-format version, and another that lists the W3C fields included in each record.
- Substitute URL-encoded equivalents for spaces and non-standard characters in field values.

  These non-standard characters consist of all ASCII codes below 32 and above 127, plus the characters in the following table. The URL encoding standard is RFC 1738. For more information, go to http://www.ietf.org/rfc/rfc1738.txt.

URL-Encoded Value	Character
%3C	<
%3E	>
%22	"
%23	#
%25	%
%7B	{
%7D	}
%7C	|
%5C	\
%5E	^
%7E	~
%5B	[
%5D	]
%60	`
%27	'
%20	space

Web Distribution Log File Format

The log file for a web distribution includes the following fields in the listed order.

Field	Description
date	The date (UTC) on which the event occurred, for example, 2009-03-10.

Field	Description
time	Time when the server finished processing the request (UTC), for example, 01:42:39.
x-edge-location	The edge location that served the request. Each edge location is identified by a three-letter code and an arbitrarily assigned number, for example, DFW3. The three-letter code typically corresponds with the International Air Transport Association airport code for an airport near the edge location. (These abbreviations may change in the future.) For a list of edge locations, see the Amazon CloudFront detail page, http://aws.amazon.com/cloudfront.
sc-bytes	Server to client bytes, including headers, for example, 1045619.
c-ip	Client IP, for example, 192.0.2.183.
cs-method	HTTP access method.
cs(Host)	DNS name (the CloudFront distribution name specified in the request). If you made the request to a CNAME, the DNS name field will contain the underlying distribution DNS name, not the CNAME.
cs-uri-stem	URI stem (for example, /images/daily-ad.jpg).
sc-status	One of the following values: • An HTTP status code (for example, `200`). For more information, see HTTP 4xx and 5xx Status Codes that CloudFront Caches (p. 117). • `000`, which indicates that the viewer closed the connection (for example, closed the browser tab) before CloudFront could respond to a request.
cs(Referer)	The referrer.
cs(User-Agent)	The user agent.
cs-uri-query	The query string portion of the URI that is included on the connect string. When a URI doesn't contain a query string, the log file contains a single hyphen (-) in the `cs-uri-query` field for that request. The encoding standard is RFC 1738. For more information, see Log File Format (p. 185).
cs(Cookie)	The cookie header in the request, including name-value pairs and the associated attributes. If you enable cookie logging, CloudFront logs the cookies in all requests regardless of which cookies you choose to forward to the origin: none, all, or a whitelist of cookie names. When a request doesn't include a cookie header, the log file contains a single hyphen (-) in the `cs(Cookie)` field for that request. For more information about cookies, see Configuring CloudFront to Cache Objects Based on Cookies (p. 76).

Field	Description
x-edge-result-type	The result type of a request. Result types include: • `Hit`: CloudFront served the object to the viewer from the edge cache. • `RefreshHit`: CloudFront found the object in the edge cache but it had expired, so CloudFront contacted the origin to verify that the cache has the latest version of the object. • `Miss`: The request could not be satisfied by an object in the edge cache, so CloudFront forwarded the request to the origin server and returned the result to the viewer. • `LimitExceeded`: The request was denied because a CloudFront limit was exceeded. • `CapacityExceeded`: CloudFront returned a 503 error because the edge location didn't have enough capacity at the time of the request to serve the object. • `Error`: Typically, this means the request resulted in a client error (`sc-status` is `4`*xx*) or a server error (`sc-status` is `5`*xx*). If `sc-status` is `403` and you configured CloudFront to restrict the geographic distribution of your content, the request might have come from a restricted location. For more information about geo restriction, see Restricting the Geographic Distribution of Your Content (p. 56). If `sc-status` is `2`*xx*, the client probably disconnected before finishing the download.
x-edge-request-id	An encrypted string that uniquely identifies a request.
x-host-header	The value that the viewer included in the Host header for this request. This is the domain name in the request: • If you're using the CloudFront domain name (http://d111111abcdef8.cloudfront.net/logo.png), the `x-host-header` column contains the domain name that CloudFront assigned to your distribution, such as `d111111abcdef8.cloudfront.net`. • If you're using alternate domain names (http://example.com/logo.png), the `x-host-header` column contains the alternate domain name, such as `example.com`. To use alternate domain names, you must add them to your distribution. For more information, see Using Alternate Domain Names (CNAMEs) (p. 29).
cs-protocol	The protocol the viewer specified in the request, either `http` or `https`.
cs-bytes	The number of bytes of data that the viewer included in the request (client to server bytes), including headers.
time-taken	The number of seconds between the time a CloudFront edge server receives a viewer's request and the time that CloudFront writes the last byte of the response to the server's output queue as measured on the server. From the perspective of the viewer, the total time to get the full object will be longer than this value due to network latency and TCP buffering.

> **Note**
> Question marks (?) in URLs and query strings are not included in the log.

The following is an example log file for a web distribution.

```
#Version: 1.0
#Fields: date time x-edge-location sc-bytes c-ip cs-method cs(Host) cs-uri-stem
 sc-status cs(Referer) cs(User-Agent) cs-uri-query cs(Cookie) x-edge-result-
type x-edge-request-id x-host-header cs-protocol cs-bytes time-taken
05/01/2014 01:13:11 FRA2 182 192.0.2.10 GET d111111abcdef8.cloudfront.net
/view/my/file.html 200 www.displaymyfiles.com Mozilla/4.0%20(compat
ible;%20MSIE%205.0b1;%20Mac_PowerPC) - zip=98101 RefreshHit MRVMF7KydIvxMWf
JIglgwHQwZsbG2IhRJ07sn9AkKUFSHS9EXAMPLE== d111111abcdef8.cloudfront.net http -
 0.001
05/01/2014 01:13:12 LAX1 2390282 192.0.2.202 GET d111111abcdef8.cloudfront.net
 /soundtrack/happy.mp3 304 www.unknownsingers.com Mozilla/4.0%20(compat
ible;%20MSIE%207.0;%20Windows%20NT%205.1) a=b&c=d zip=50158 Hit
xGN7KWpVEmB9Dp7ctcVFQC4E-nrcOcEKS3QyAez--06dV7TEXAMPLE== d111111abcdef8.cloud
front.net http - 0.002
```

RTMP Distribution Log File Format

Each record in an RTMP access log represents a playback event, for example, connect, play, pause, stop, disconnect, and so on. As a result, CloudFront generates multiple log records each time a viewer watches a video. To relate log records that stem from the same stream ID, use the *x-sid* field.

> **Note**
> Some fields have values for all events, and some have values only for Play, Stop, Pause, Unpause, and Seek events. Usually, when the log file contains a single hyphen (-) for a field, the field isn't relevant for the corresponding event.

The following table describes the fields that are present in each record in the RTMP distribution log file, regardless of the type of event. The fields appear in the log in the order listed.

Field	Description
date	Date (UTC) on which the event occurred.
time	Time when the server received the request (UTC), for example, 01:42:39.
x-edge-location	The edge location where the playback event occurred. Each edge location is identified by a three-letter code and an arbitrarily assigned number, for example, DFW3. The three-letter code typically corresponds with the International Air Transport Association airport code for an airport near the edge location. (These abbreviations may change in the future.) For a list of edge locations, see the Amazon CloudFront detail page, http://aws.amazon.com/cloudfront.
c-ip	Client IP, for example, 192.0.2.183.
x-event	The event type. This is a Connect, Disconnect, Play, Stop, Pause, Unpause, or Seek event.
sc-bytes	The running total number of bytes sent from the server to the client, up to the time of the event.
x-cf-status	A code indicating the status of the event. Currently, "OK" is the only value for this field. New functionality in the future could require new status codes.

Field	Description
x-cf-client-id	An opaque string identifier that can be used to differentiate clients This value is unique for each connection.
cs-uri-stem	The stem portion of the URI, including the application and the application instance. This is sometimes referred to as the FMS *connect string*. For example, rtmp://shqshne4jdp4b6.cloudfront.net/cfx/st.
cs-uri-query	The query string portion of the URI that is included on the connect string.
c-referrer	The URI of the referrer.
x-page-url	The URL of the page from which the SWF is linked.
c-user-agent	The user agent.

The following fields usually have values only for Play, Stop, Pause, Unpause, and Seek events. For other events, they contain a single hyphen (-). These fields appear in the log after the fields in the previous table and in the order listed.

Field	Description
x-sname	The stream name.
x-sname-query	The stream query string, if any.
x-file-ext	The stream type, for instance, FLV.
x-sid	The stream ID. This is a unique integer identifier for the connection.

Note
Question marks (?) in URLs and query strings are not included in the log.

The following is an example of a log file for an RTMP distribution.

```
#Version: 1.0
#Fields: date time x-edge-location c-ip x-event sc-bytes x-cf-status x-cf-client-
id cs-uri-stem cs-uri-query c-referrer x-page-url  c-user-agent x-sname x-sname-
query x-file-ext x-sid
2010-03-12   23:51:20   SEA4   192.0.2.147   connect   2014   OK
bfd8a98bee0840d9b871b7f6ade9908f   rtmp://shqshne4jdp4b6.cloudfront.net/cfx/st
  key=value   http://player.longtailvideo.com/player.swf   http://www.long
tailvideo.com/support/jw-player-setup-wizard?example=204   LNX%2010,0,32,18
-   -   -   -
2010-03-12   23:51:21   SEA4   192.0.2.222   play   3914   OK
bfd8a98bee0840d9b871b7f6ade9908f   rtmp://shqshne4jdp4b6.cloudfront.net/cfx/st
  key=value   http://player.longtailvideo.com/player.swf   http://www.long
tailvideo.com/support/jw-player-setup-wizard?example=204   LNX%2010,0,32,18
myvideo   p=2&q=4   flv   1
2010-03-12   23:53:44   SEA4   192.0.2.4   stop   323914   OK
bfd8a98bee0840d9b871b7f6ade9908f   rtmp://shqshne4jdp4b6.cloudfront.net/cfx/st
  key=value   http://player.longtailvideo.com/player.swf   http://www.long
tailvideo.com/support/jw-player-setup-wizard?example=204   LNX%2010,0,32,18
dir/other/myvideo   p=2&q=4   flv   1
2010-03-12   23:53:44   SEA4   192.0.2.103   play   8783724   OK
```

```
bfd8a98bee0840d9b871b7f6ade9908f   rtmp://shqshne4jdp4b6.cloudfront.net/cfx/st
  key=value   http://player.longtailvideo.com/player.swf   http://www.long
tailvideo.com/support/jw-player-setup-wizard?example=204   LNX%2010,0,32,18
dir/favs/myothervideo   p=42&q=14   mp4   2
2010-03-12   23:56:21   SEA4   192.0.2.199   stop   429822014   OK
bfd8a98bee0840d9b871b7f6ade9908f   rtmp://shqshne4jdp4b6.cloudfront.net/cfx/st
  key=value   http://player.longtailvideo.com/player.swf   http://www.long
tailvideo.com/support/jw-player-setup-wizard?example=204   LNX%2010,0,32,18
dir/favs/myothervideo   p=42&q=14   mp4   2
2010-03-12   23:59:44   SEA4   192.0.2.14   disconnect   429824092   OK
bfd8a98bee0840d9b871b7f6ade9908f   rtmp://shqshne4jdp4b6.cloudfront.net/cfx/st
  key=value   http://player.longtailvideo.com/player.swf   http://www.long
tailvideo.com/support/jw-player-setup-wizard?example=204   LNX%2010,0,32,18
-   -   -   -
```

Charges for Access Logs

Access logging is an optional feature of CloudFront. There is no extra charge for enabling access logging. However, you accrue the usual Amazon S3 charges for storing and accessing the files on Amazon S3 (you can delete them at any time). For more information about charges for CloudFront see CloudFront Billing and Usage Reports (p. 7).

Using AWS CloudTrail to Capture Requests Sent to the CloudFront API

CloudFront is integrated with CloudTrail, an AWS service that captures information about every request that is sent to the CloudFront API by your AWS account, including your IAM users. CloudTrail periodically saves log files of these requests to an Amazon S3 bucket that you specify. CloudTrail captures information about all requests, whether they were made using the CloudFront console, the CloudFront API, the AWS SDKs, the CloudFront CLI, or another service, for example, AWS CloudFormation.

You can use information in the CloudTrail log files to determine which requests were made to CloudFront, the source IP address from which each request was made, who made the request, when it was made, and so on. To learn more about CloudTrail, including how to configure and enable it, see the *AWS CloudTrail User Guide*.

Topics

- CloudFront Information in CloudTrail Log Files (p. 192)
- Understanding CloudFront Log File Entries (p. 193)

CloudFront Information in CloudTrail Log Files

When you enable CloudTrail, CloudTrail captures every request that you make to every AWS service that CloudTrail supports. (For a list of supported services, see Supported Services in the *AWS CloudTrail User Guide*.) The log files aren't organized or sorted by service; each log file might contain records from more than one service. CloudTrail determines when to create a new log file.

> **Note**
> CloudTrail supports all CloudFront API actions.

Every log file entry contains information about who made the request. The user identity information in the log file helps you determine whether the request was made using root or IAM user credentials, using temporary security credentials for a role or federated user, or by another AWS service. For more information, see userIdentity Element in the *AWS CloudTrail User Guide*.

You can store log files for as long as you want. You can also define Amazon S3 lifecycle rules to archive or delete log files automatically.

By default, your log files are encrypted by using Amazon S3 server-side encryption (SSE).

You can choose to have CloudTrail publish Amazon SNS notifications when new log files are delivered if you want to take quick action upon log file delivery. For more information, see Configuring Amazon SNS Notifications in the *AWS CloudTrail User Guide*.

You can also aggregate log files from multiple AWS regions and multiple AWS accounts into a single Amazon SNS bucket. For more information, see Aggregating CloudTrail Log Files to a Single Amazon S3 Bucket in the *AWS CloudTrail User Guide*.

Understanding CloudFront Log File Entries

Each JSON-formatted CloudTrail log file can contain one or more log entries. A log entry represents a single request from any source and includes information about the requested action, including any parameters, the date and time of the action, and so on. The log entries are not guaranteed to be in any particular order; they are not an ordered stack trace of API calls.

The `eventName` element identifies the action that occurred and the API version that was used to perform that action. For example, the following `eventName` value indicates that a web distribution was updated, and the 2014-01-31 API version was used to perform the action:

```
UpdateDistribution2014_01_31
```

The following example shows a CloudTrail log entry that demonstrates five actions:

- Updating a web distribution configuration. The value of `eventName` is `UpdateDistribution`.
- Listing web distributions that are associated with the current account. The value of `eventName` is `ListDistributions`.
- Getting the configuration for a specific web distribution. The value of `eventName` is `GetDistribution`.
- Creating an invalidation batch request. The value of `eventName` is `CreateInvalidation`.
- Listing origin access identities that are associated with the current account. The value of `eventName` is `ListCloudFrontOriginAccessIdentities`.

```
{
   "Records": [{
      "eventVersion": "1.01",
      "userIdentity": {
         "type": "IAMUser",
         "principalId": "A1B2C3D4E5F6G7EXAMPLE",
         "arn": "arn:aws:iam::111122223333:user/smithj",
         "accountId": "111122223333",
         "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
         "userName": "smithj"
      },
      "eventTime": "2014-05-06T18:00:32Z",
      "eventName": "UpdateDistribution2014_01_31",
      "sourceIPAddress": "192.0.2.17",
      "userAgent": "aws-sdk-ruby/1.39.0 ruby/1.9.3 x86_64-linux",
      "requestParameters": {
         "id": "EDFDVBD6EXAMPLE",
         "ifMatch": "E9LHASXEXAMPLE",
         "distributionConfig": {
            "restrictions": {
               "geoRestriction": {
                  "quantity": 0,
                  "restrictionType": "none"
               }
            },
            "customErrorResponses": {
               "quantity": 0
            },
            "defaultRootObject": "index.html",
            "aliases": {
               "quantity": 1,
               "items": ["example.com"]
            },
            "logging": {
               "bucket": "",
               "enabled": false,
               "prefix": "",
               "includeCookies": false
            },
            "viewerCertificate": {
               "iAMCertificateId": "A1B2C3D4E5F6G7EXAMPLE",
               "sSLSupportMethod": "sni-only"
            },
            "callerReference": "2014-05-06 64832",
            "defaultCacheBehavior": {
               "targetOriginId": "Images",
               "allowedMethods": {
                  "items": ["GET",
                  "HEAD"],
                  "quantity": 2
               },
               "forwardedValues": {
                  "cookies": {
                     "forward": "none"
                  },
                  "queryString": false
               },
               "minTTL": 300,
               "trustedSigners": {
                  "enabled": false,
                  "quantity": 0
               },
               "viewerProtocolPolicy": "redirect-to-https",
               "smoothStreaming": false
            },
            "origins": {
               "items": [{
                  "customOriginConfig": {
                     "hTTPSPort": 443,
                     "originProtocolPolicy": "http-only",
                     "hTTPPort": 80
                  },
                "domainName": "myawsbucket.s3-website-us-east-1.amazonaws.com",

                  "id": "Web page origin"
               },
               {
                  "customOriginConfig": {
                     "hTTPSPort": 443,
                     "originProtocolPolicy": "http-only",
                     "hTTPPort": 80
                  },
                  "domainName": "myotherawsbucket.s3-website-us-west-2.amazon
aws.com",
                  "id": "Images"
               }],
               "quantity": 2
            },
            "enabled": true,
            "cacheBehaviors": {
                  "allowedMethods": {
                     "items": ["GET",
                     "HEAD"],
                     "quantity": 2
                  },
                  "trustedSigners": {
                     "enabled": false,
                     "quantity": 0
                  },
                  "targetOriginId": "Web page origin",
                  "smoothStreaming": false,
                  "viewerProtocolPolicy": "redirect-to-https",
                  "minTTL": 300,
                  "forwardedValues": {
                     "cookies": {
                        "forward": "none"
                     },
                     "queryString": false
                  },
                  "pathPattern": "*.html"
               }],
               "quantity": 1
            },
            "priceClass": "PriceClass_All",
            "comment": "Added an origin and a cache behavior"
         }
      },
      "responseElements": {
         "eTag": "E2QWRUHEXAMPLE",
         "distribution": {
            "domainName": "d111111abcdef8.cloudfront.net",
            "status": "InProgress",
            "distributionConfig": {
            distributionConfig response omitted
            },
            "id": "EDFDVBD6EXAMPLE",
            "lastModifiedTime": "May 6, 2014 6:00:32 PM",
            "activeTrustedSigners": {
               "quantity": 0,
               "enabled": false
            },
            "inProgressInvalidationBatches": 0
         }
      },
      "requestID": "4e6b66f9-d548-11e3-a8a9-73e33example",
      "eventID": "5ab02562-0fc5-43d0-b7b6-90293example"
   },
   {
      "eventVersion": "1.01",
      "userIdentity": {
         "type": "IAMUser",
         "principalId": "A1B2C3D4E5F6G7EXAMPLE",
         "arn": "arn:aws:iam::111122223333:user/smithj",
         "accountId": "111122223333",
         "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
         "userName": "smithj"
      },
      "eventTime": "2014-05-06T18:01:35Z",
      "eventName": "ListDistributions2014_01_31",
      "sourceIPAddress": "192.0.2.17",
      "userAgent": "aws-sdk-ruby/1.39.0 ruby/1.9.3 x86_64-linux",
      "requestParameters": null,
      "responseElements": null,
      "requestID": "52de9f97-d548-11e3-8fb9-4dad0example",
      "eventID": "eb91f423-6dd3-4bb0-a148-3cdfbexample"
   },
   {
      "eventVersion": "1.01",
      "userIdentity": {
         "type": "IAMUser",
         "principalId": "A1B2C3D4E5F6G7EXAMPLE",
         "arn": "arn:aws:iam::111122223333:user/smithj",
         "accountId": "111122223333",
         "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
         "userName": "smithj"
      },
      "eventTime": "2014-05-06T18:01:59Z",
      "eventName": "GetDistribution2014_01_31",
      "sourceIPAddress": "192.0.2.17",
      "userAgent": "aws-sdk-ruby/1.39.0 ruby/1.9.3 x86_64-linux",
      "requestParameters": {
         "id": "EDFDVBD6EXAMPLE"
      },
      "responseElements": null,
      "requestID": "497b3622-d548-11e3-8fb9-4dad0example",
      "eventID": "c32289c7-005a-46f7-9801-cba41example"
   },
   {
      "eventVersion": "1.01",
      "userIdentity": {
         "type": "IAMUser",
         "principalId": "A1B2C3D4E5F6G7EXAMPLE",
         "arn": "arn:aws:iam::111122223333:user/smithj",
         "accountId": "111122223333",
         "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
         "userName": "smithj"
      },
      "eventTime": "2014-05-06T18:02:27Z",
      "eventName": "CreateInvalidation2014_01_31",
      "sourceIPAddress": "192.0.2.17",
      "userAgent": "aws-sdk-ruby/1.39.0 ruby/1.9.3 x86_64-linux",
      "requestParameters": {
         "invalidationBatch": {
            "callerReference": "2014-05-06 64947",
            "paths": {
               "quantity": 3,
               "items": ["/images/new.jpg",
               "/images/logo.jpg",
               "/images/banner.jpg"]
            }
         },
         "distributionId": "EDFDVBD6EXAMPLE"
      },
      "responseElements": {
         "invalidation": {
            "createTime": "May 6, 2014 6:02:27 PM",
            "invalidationBatch": {
               "callerReference": "2014-05-06 64947",
               "paths": {
                  "quantity": 3,
                  "items": ["/images/banner.jpg",
                  "/images/logo.jpg",
                  "/images/new.jpg"]
               }
            },
            "status": "InProgress",
            "id": "ISRZ85EXAMPLE"
         },
        "location": "https://cloudfront.amazonaws.com/2014-01-31/distribution/ED
FDVBD6EXAMPLE/invalidation/ISRZ85EXAMPLE"
      },
      "requestID": "4e200613-d548-11e3-a8a9-73e33example",
      "eventID": "191ebb93-66b7-4517-a741-92b0eexample"
   },
   {
      "eventVersion": "1.01",
      "userIdentity": {
         "type": "IAMUser",
         "principalId": "A1B2C3D4E5F6G7EXAMPLE",
         "arn": "arn:aws:iam::111122223333:user/smithj",
         "accountId": "111122223333",
         "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
         "userName": "smithj"
      },
      "eventTime": "2014-05-06T18:03:08Z",
      "eventName": "ListCloudFrontOriginAccessIdentities2014_01_31",
      "sourceIPAddress": "192.0.2.17",
      "userAgent": "aws-sdk-ruby/1.39.0 ruby/1.9.3 x86_64-linux",
      "requestParameters": null,
      "responseElements": null,
      "requestID": "42ca4299-d548-11e3-8fb9-4dad0example",
      "eventID": "7aeb434f-eb55-4e2a-82d8-417d5example"
   }]
}
```

Troubleshooting

Topics

- I can't view the files in my web distribution. (p. 198)
- I can't view the files in my RTMP distribution. (p. 200)
- Error Message: Certificate: <certificate-id> is being used by CloudFront. (p. 200)

I can't view the files in my web distribution.

If you cannot view file in your CloudFront web distribution, the following topics describe some common solutions.

Did you sign up for both CloudFront and Amazon S3?

To use Amazon CloudFront with an Amazon S3 origin, you must sign up for both CloudFront and Amazon S3, separately. For more information about signing up for CloudFront and Amazon S3, see Getting Started with CloudFront (p. 17).

Are your Amazon S3 bucket and object permissions set correctly?

If you are using CloudFront with an Amazon S3 origin, the original versions of your content are stored in an Amazon S3 bucket. The easiest way to use CloudFront with Amazon S3 is to make all your objects publicly readable in Amazon S3. To do this, you must explicitly enable public read privileges for each object you upload to Amazon S3.

If your content is not publicly readable, you need to create a CloudFront origin access identity so CloudFront can access it. For more information about CloudFront origin access identities, see Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content (p. 123).

Object properties and bucket properties are independent. You must explicitly grant privileges to each object in Amazon S3. Objects do not inherit properties from buckets and object properties must be set independently of the bucket.

Is your alternate domain name (CNAME) correctly configured?

If you already have an existing CNAME record for your domain name, update that record or replace it with a new one that points to your distribution's domain name.

Also, make sure your CNAME record points to your distribution's domain name, not your Amazon S3 bucket. You can confirm that the CNAME record in your DNS system points to your distribution's domain name. To do so, use a DNS tool like dig. (For information about dig, go to http://www.kloth.net/services/dig.php.)

The following shows an example dig request on a domain name called `images.example.com`, and the relevant part of the response. Under `ANSWER SECTION`, see the line that contains `CNAME`. The CNAME record for your domain name is set up correctly if the value on the right side of CNAME is your CloudFront distribution's domain name. If it's your Amazon S3 origin server bucket or some other domain name, then the CNAME record is set up incorrectly.

```
[prompt]> dig images.example.com

; <<> DiG 9.3.3rc2 <<> images.example.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15917
;; flags: qr rd ra; QUERY: 1, ANSWER: 9, AUTHORITY: 2, ADDITIONAL: 0
;; QUESTION SECTION:
;images.example.com.     IN    A
;; ANSWER SECTION:
images.example.com. 10800 IN  CNAME  d111111abcdef8.cloudfront.net.
...
...
```

For more information about CNAMEs, see Using Alternate Domain Names (CNAMEs) (p. 29).

Are you referencing the correct URL for your CloudFront distribution?

Make sure the URL you're referencing uses your CloudFront distribution domain name (or your CNAME), not your Amazon S3 bucket or custom origin.

Do you need help troubleshooting a custom origin?

If you need AWS to help you troubleshoot a custom origin, we will probably need to inspect the `X-Amz-Cf-Id` header entries from your requests. If you are not already logging these entries, you might want to consider it for the future. For more information, see Requirements and Recommendations for Using Amazon EC2 and Other Custom Origins (p. 56).

I can't view the files in my RTMP distribution.

If you cannot view the files in an RTMP distribution, are your URL and your playback client correctly configured? RTMP distributions require you to use an RTMP protocol instead of HTTP, and you must make a few minor configuration changes to your playback client. For information about creating RTMP distributions, see Task List for Streaming Media Files Using RTMP (p. 62).

Error Message: Certificate: <certificate-id> is being used by CloudFront.

Problem: You're trying to delete an SSL certificate from the IAM certificate store, and you're getting the message "Certificate: <certificate-id> is being used by CloudFront."

Solution: Every CloudFront web distribution must be associated either with the default CloudFront certificate or with a custom SSL certificate. Before you can delete an SSL certificate, you need to either rotate SSL certificates (replace the current custom SSL certificate with another custom SSL certificate) or revert from using a custom SSL certificate to using the default CloudFront certificate. Perform the procedure in the applicable section:

- Rotating SSL Certificates (p. 175)
- Reverting from a Custom SSL Certificate to the Default CloudFront Certificate (p. 176)

Load Testing CloudFront

Traditional load testing methods don't work well with CloudFront because CloudFront uses DNS to balance loads across geographically dispersed edge locations and within each edge location. When a client requests content from CloudFront, the client receives a DNS response that includes a set of IP addresses. If you test by sending requests to just one of the IP addresses that DNS returns, you're testing only a small subset of the resources in one CloudFront edge location, which doesn't accurately represent actual traffic patterns. Depending on the volume of data requested, testing in this way may overload and degrade the performance of that small subset of CloudFront servers.

CloudFront is designed to scale for viewers that have different client IP addresses and different DNS resolvers across multiple geographic regions. To perform load testing that accurately assesses CloudFront performance, we recommend that you do all of the following:

- Send client requests from multiple geographic regions.
- Configure your test so each client makes an independent DNS request; each client will then receive a different set of IP addresses from DNS.
- For each client that is making requests, spread your client requests across the set of IP addresses that are returned by DNS, which ensures that the load is distributed across multiple servers in a CloudFront edge location.

CloudFront Tutorials

The following tutorials explain how to use CloudFront for live streaming, for geoblocking, and for RTMP streaming.

Live Streaming

- Live HTTP Streaming Using CloudFront and Adobe Media Server 5.0 (p. 202)
- Live Smooth Streaming Using Amazon CloudFront and IIS Media Services 4.1 (p. 220)
- Live HTTP Streaming Using Wowza Streaming Engine 4.0 (p. 235)

Geoblocking

- Restricting Access to Files in a CloudFront Distribution Based on Geographic Location (Geoblocking) (p. 244)

RTMP Streaming

- On-Demand Video Streaming Using CloudFront and Adobe Flash Player (p. 269)
- On-Demand Video Streaming Using CloudFront and Flowplayer for Adobe Flash (p. 274)
- On-Demand Video Streaming Using CloudFront and JW Player (p. 279)

Live HTTP Streaming Using CloudFront and Adobe Media Server 5.0

Topics

- Overview (p. 203)
- Steps to Configure Live Streaming (p. 204)
- Creating an Amazon Web Services Account (p. 204)

- Creating an Amazon EC2 Key Pair (p. 204)
- Subscribing to Adobe Media Server (p. 205)
- Creating an AWS CloudFormation Stack for Live Streaming (p. 206)
- Verifying that Adobe Media Server Is Running (p. 207)
- Setting Up Adobe Flash Media Live Encoder to Publish a Live Stream (p. 208)
- Embedding Strobe Media Playback for an Amazon CloudFront Live HTTP Stream in a Web Application (p. 211)
- Deleting an AWS CloudFormation Stack and an Amazon EBS Volume for Live Streaming (p. 212)
- Frequently Asked Questions (p. 213)
- Additional Documentation (p. 219)

With Amazon Web Services live streaming, you can use Adobe Media Server version 5.0 to stream live performances, webinars, and other events. This tutorial walks you through the process of configuring live streaming with Adobe Media Server 5.0.

Overview

Adobe Media Server 5.0 supports two HTTP streaming formats:

- HLS (HTTP Live Streaming), supported by iOS devices
- HDS (HTTP Dynamic Streaming), supported by Flash applications (including Strobe Media Playback)

> **Note**
> An earlier version of this tutorial explained how to configure HDS streaming using the Flash Media Playback player, but Adobe stopped supporting that player. We've updated the tutorial to use Strobe Media Playback, an open-source media player that is similar in functionality to Flash Media Playback.

Here's how Adobe Media Server and CloudFront work together to stream an event in realtime:

1. You use AWS CloudFormation to provision an Amazon EC2 instance running Adobe Media Server 5.0 and to create a CloudFront distribution, as described in this tutorial.
2. You capture your event using a digital video camera, for example, the video camera in a laptop computer.
3. You use an encoder on the site of the event, for example, Adobe Flash Media Live Encoder, to compress the raw video feed and send it to Adobe Media Server. (Flash Media Live Encoder is a free download and is available for Windows and Mac OS.)
4. Adobe Media Server breaks the video stream into a series of smaller files. This server is the origin for your CloudFront distribution.
5. When your users browse to the CloudFront URL that you gave them to view the event, CloudFront routes their HTTP requests to the nearest edge location (by latency).
6. The edge location requests the video stream from Adobe Media Server.
7. Adobe Media Server returns the video stream in small files to the CloudFront edge location.
8. The CloudFront edge location serves the video stream to the viewer that made the request, and caches the small files to speed up the response to subsequent requests for the live stream.

This tutorial summarizes how to integrate CloudFront with Adobe Media Server running on an Amazon EC2 instance. For more information about Adobe Media Server and about the AWS services that you use for live streaming, see the following:

- For more information about Adobe Media Server options not covered in this tutorial, see Additional Documentation (p. 219).

- For information about available Adobe Media Server features, see Adobe Media Server 5 on Amazon Web Services.
- To review the new features in Adobe Media Server 5.0, see What's New in Adobe Media Server 5.0.1 on the Adobe website.
- For more information about how to manage and secure your Amazon EC2 instance, see the Amazon EC2 documentation.
- For more information about AWS CloudFormation, see AWS CloudFormation Documentation.
- For more help, see Frequently Asked Questions (p. 213).

Steps to Configure Live Streaming

To set up live streaming with Amazon Web Services (AWS), review the system requirements for Adobe Flash Player. Then perform the procedures in the following sections:

1. Creating an Amazon Web Services Account (p. 204)
2. Creating an Amazon EC2 Key Pair (p. 204)
3. Subscribing to Adobe Media Server (p. 205)
4. Creating an AWS CloudFormation Stack for Live Streaming (p. 206)
5. Verifying that Adobe Media Server Is Running (p. 207)
6. Setting Up Adobe Flash Media Live Encoder to Publish a Live Stream (p. 208)
7. Embedding Strobe Media Playback for an Amazon CloudFront Live HTTP Stream in a Web Application (p. 211)
8. Deleting an AWS CloudFormation Stack and an Amazon EBS Volume for Live Streaming (p. 212)

Creating an Amazon Web Services Account

If you already have an AWS account, skip to Creating an Amazon EC2 Key Pair (p. 204). If you don't already have an AWS account, use the following procedure to create one.

> **Note**
> When you create an account, AWS automatically signs up the account for all services. You are charged only for the services you use.

To create an AWS account

1. Go to http://aws.amazon.com, and click **Create an AWS Account**.
2. Follow the on-screen instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.

Next: Creating an Amazon EC2 Key Pair (p. 204)

Creating an Amazon EC2 Key Pair

If you already have an Amazon EC2 key pair in the Amazon EC2 region in which you want to configure live streaming, skip to Subscribing to Adobe Media Server (p. 205). If you don't have a key pair in that region, perform the following procedure.

A key pair is a security credential similar to a password. You specify a key pair when you create an AWS CloudFormation stack for live streaming, later in this process. After live streaming is configured, you use the key pair to securely connect to your Amazon EC2 instance.

To create an Amazon EC2 key pair

1. Sign in to the AWS Management Console and open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the **Region** list, click the region in which you want to create the key pair.

   You must create the key pair in the same region in which you will create your AWS CloudFormation stack for live streaming later in this process. We recommend that you create the key pair and the stack for live streaming in the region that is closest to the users who will be doing the streaming.
3. In the navigation pane, click **Key Pairs**.
4. In the **Key Pairs** pane, click **Create Key Pair**.
5. In the Create Key Pair dialog box, enter a name for the key pair, and make note of the name. You'll enter this value when you create an AWS CloudFormation live-streaming stack, later in the process of setting up live streaming.
6. Click **Create**.
7. In the Opening <key_pair_name>.pem dialog box, save the .pem file to a safe place on your computer.

   > **Important**
   > This is the only opportunity you'll have to download and save your private key.
8. Click **Close** to close the Create Key Pair dialog box.

Next: Subscribing to Adobe Media Server (p. 205)

Subscribing to Adobe Media Server

Perform the following procedure to subscribe to Adobe Media Server for Amazon Web Services through AWS Marketplace.

> **Important**
> You can subscribe an AWS account to Adobe Media Server only once. If your AWS account already has an Adobe Media Server subscription, use that subscription to configure live streaming.

Adobe Media Server has a $5.00 monthly subscription fee, which includes an unlimited number of Adobe Media Server instances. In addition to the monthly subscription fee, you pay a fee for hourly usage and a fee for data transfer. You can view a detailed price list as part of the following procedure.

> **Note**
> In an earlier version of this tutorial, you subscribed to Adobe Media Server using Amazon DevPay instead of AWS Marketplace. If you're currently running Adobe Media Server and you subscribed using Amazon DevPay, go to the Adobe Media Server on Amazon Web Services page and cancel your Amazon DevPay subscription to Adobe Media Server. Otherwise, you'll continue to be billed $5 per month for the old subscription through Amazon DevPay, and you'll be billed an additional $5 per month for the new subscription through AWS Marketplace.

To order Adobe Media Server 5 for Amazon Web Services

1. Go to the Adobe Media Server 5 Extended page on the AWS Marketplace website.
2. Review product information and click **Continue**.
3. On the **Launch on EC2: Adobe Media Server 5 Extended** page, click the **Manual Launch with EC2 Console, APIs or CLI** tab.
4. In the **Pricing Details** section, select the region in which to create an Amazon EC2 instance for live streaming. Review the corresponding pricing information.

> **Important**
> Don't use the buttons on this page to launch Adobe Media Server. In the next procedure, you create an AWS CloudFormation stack that launches an Amazon EC2 instance and installs Adobe Media Server.

5. Click **Accept Terms** to sign up for a monthly subscription.

Next: Creating an AWS CloudFormation Stack for Live Streaming (p. 206)

Creating an AWS CloudFormation Stack for Live Streaming

The following procedure uses an AWS CloudFormation template to create a stack that launches the AWS resources required by live streaming, including an Amazon EC2 instance and a CloudFront distribution.

> **Important**
> You incur hourly charges for an Amazon EC2 instance beginning when you create the AWS CloudFormation stack that deploys the Amazon EC2 instance. Charges continue to accrue until you delete the AWS CloudFormation stack regardless of whether you use the Amazon EC2 instance to stream live video. For more information, see the Adobe Media Server 5 Extended page on the AWS Marketplace website. When your live event is over, delete the stack that you created for live streaming. This deletes the AWS resources that were created for your live-streaming event, and stops the AWS charges for the resources. For more information, see Deleting an AWS CloudFormation Stack and an Amazon EBS Volume for Live Streaming (p. 212).

For more information about AWS CloudFormation, see AWS CloudFormation Documentation.

To create an AWS CloudFormation stack for live streaming

1. To start the Create Stack wizard, click one of the following Amazon EC2 regions:

   - Create a stack in US East (Virginia)
   - Create a stack in US West (Oregon)
   - Create a stack in US West (Northern California)
   - Create a stack in EU (Ireland)
   - Create a stack in Asia Pacific (Singapore)
   - Create a stack in Asia Pacific (Tokyo)
   - Create a stack in Asia Pacific (Australia)
   - Create a stack in South America (Sao Paulo)

   The wizard starts and the applicable URL automatically appears in the **Provide an S3 URL to template** field.

   > **Note**
   > If you want users to view your live stream using a Flash-based player that is hosted on your own domain, see How do I update crossdomain.xml for a Flash-based stream hosted on my own domain? (p. 215).

2. If you are not already signed in to the AWS Management Console, sign in when prompted.
3. (Optional) Change the **Stack Name**. The stack name must not contain spaces, and it must be unique within your AWS account.

   Do not change the **Template** option or the address in **Provide an S3 URL to template**.
4. Click **Next Step**.

5. On the **Specify Parameters** page, for **AMSAdminPassword**, enter a password (minimum 8 characters) for the AMS Administration Console.
6. For **AMSAdminUserName**, enter a user name. You'll use this value and the password that you entered in the previous step to log in to the AMS Administration Console after your Amazon EC2 Adobe Media Server instance is created.
7. For **InstanceType**, enter an instance type, which determines pricing for your Adobe Media Server instance. For more information about Amazon EC2 instance types, see Available Instance Types in the *Amazon Elastic Compute Cloud User Guide for Linux*.

   For information about pricing, see the Adobe Media Server 5 Extended page on the AWS Marketplace website.
8. For **KeyPair**, enter the name of an Amazon EC2 key pair in the same region that you chose in Step 1. The key pair must be associated with the account that you're currently signed in with. If you created a key pair when you performed the procedure in Creating an Amazon EC2 Key Pair (p. 204), enter the name of that key pair.
9. For **StreamName**, enter a short name (without spaces) for your live stream.
10. Click **Next Step**.
11. (Optional) On the **Add Tags** page, add one or more tags.
12. (Optional) To configure SNS notification, to specify how long you're willing to wait for the stack to be created, to choose whether to roll back changes if stack creation fails, and to enter a stack policy, click **Advanced**, and adjust settings as desired.
13. Click **Next Step**.
14. Review the settings for the stack. When you're satisfied with the settings, click **Create**, and AWS CloudFormation creates the stack.

    Creating your stack may take several minutes. To track the progress of the stack creation, select the stack, and click the **Events** tab in the bottom frame. If AWS CloudFormation cannot create the stack, the Events tab lists error messages.

    When your stack is ready, in the top frame, the status for the stack changes to **CREATE_COMPLETE**.

    When your stack is created, click the **Outputs** tab, which displays the stack creation outputs. You will use these values when you set up Adobe Flash Media Live Encoder later in the process.

Next: Verifying that Adobe Media Server Is Running (p. 207)

Verifying that Adobe Media Server Is Running

After AWS CloudFormation creates the stack, perform the following procedure to verify Adobe Media Server is running on the Amazon Amazon EC2 instance you provisioned using AWS CloudFormation.

To verify that Adobe Media Server is running

1. Open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation/.
2. In the top pane, select the stack that you created in Creating an AWS CloudFormation Stack for Live Streaming (p. 206).
3. In the bottom pane, click the **Outputs** tab.
4. Click on the value of the **AMSServer** key, which is the URL to the Amazon EC2 instance that you provisioned when you performed the procedure in Creating an AWS CloudFormation Stack for Live Streaming (p. 206).
5. The Adobe Media Server page appears and begins streaming content, which shows that Adobe Media Server is running.

If streaming does not start, return to Overview (p. 203), and verify that the values you specified in the first four tasks are correct.

If the values were all correct, but streaming still has not started, see How do I troubleshoot my Amazon EC2 instance if streaming doesn't start? (p. 219).

Next: Setting Up Adobe Flash Media Live Encoder to Publish a Live Stream (p. 208)

Setting Up Adobe Flash Media Live Encoder to Publish a Live Stream

Adobe Media Server on Amazon Web Services includes an application called livepkgr that packages published streams for delivery using HTTP Dynamic Streaming (HDS) and HTTP Live Streaming (HLS).

The following procedure shows how to set up Adobe Flash Media Live Encoder (FMLE) to publish your live stream to the livepkgr application on Adobe Media Server 5.0.

> **Note**
> The Windows version of Flash Media Live Encoder doesn't support the AAC audio format. To add support for AAC, Adobe recommends that you purchase the MainConcept AAC encoder.

To specify live-streaming settings in Flash Media Live Encoder

1. Sign in to the computer that you'll use to broadcast the live stream.
2. Open a web browser, and go to the Adobe Flash Media Live Encoder page.
3. Download and install Flash Media Live Encoder.

   > **Note**
   > Flash Media Live Encoder is free, but to download it, you need an Adobe account (also free).

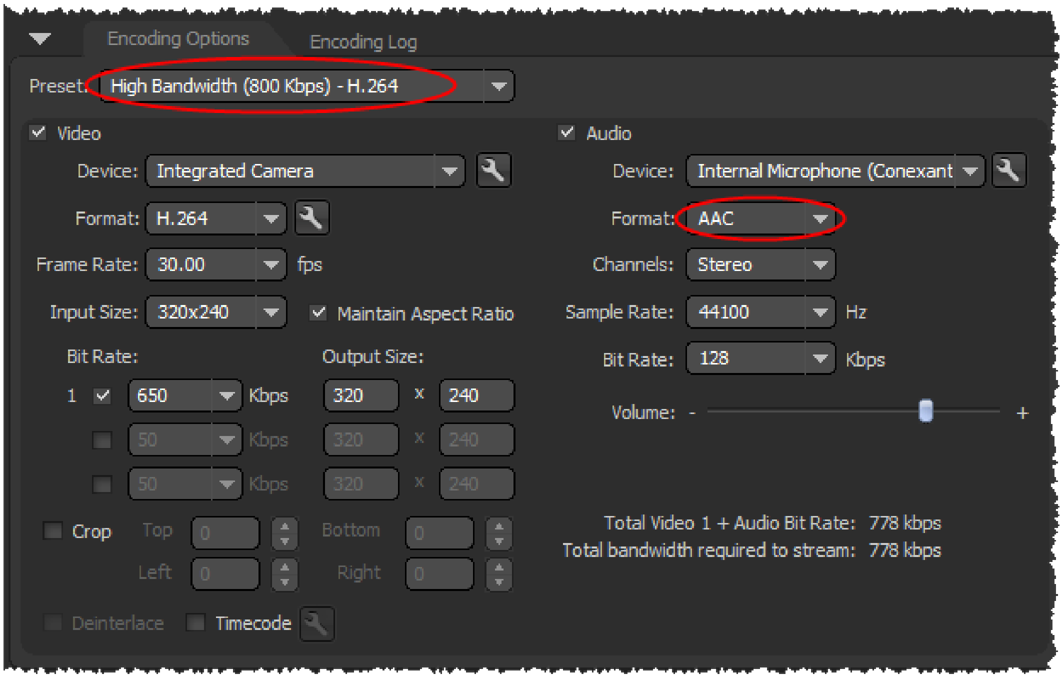
4. Open the Flash Media Live Encoder `config.xml` file in a text editor. The default installation location depends on your operating system:

   - **32-bit Windows:** `C:\Program Files\Adobe\Flash Media Live Encoder 3.2`.
   - **64-bit Windows:** `C:\Program Files (x86)\Adobe\Flash Media Live Encoder 3.2\Conf`.
   - **Macintosh:** `Applications:Adobe:Flash Media Live Encoder 3.2`.

5. In `config.xml`, set the value of the following `<enable>` element to `true`:

```
<flashmedialiveencoder_config>
   ...
   <mbrconfig>
      ...
      <streamsynchronization>
         ...
         <!-- "true" to enable this feature, "false" to disable. -->
         <enable>true</enable>
```

6. Save the file.
7. Run Flash Media Live Encoder.
8. On the **Encoding Options** tab, for **Preset**, select **High Bandwidth (800 Kbps) — H.264**.
9. On the **Encoding Options** tab, under the **Audio** check box, for **Format**, select **AAC**.



Encoding Options | Encoding Log

Preset: High Bandwidth (800 Kbps) - H.264

☑ Video

Device: Integrated Camera

Format: H.264

Frame Rate: 30.00 fps

Input Size: 320x240 ☑ Maintain Aspect Ratio

Bit Rate: / Output Size:

1 ☑ 650 Kbps 320 x 240

□ 50 Kbps 320 x 240

□ 50 Kbps 320 x 240

□ Crop Top 0 Bottom 0

Left 0 Right 0

□ Deinterlace □ Timecode

☑ Audio

Device: Internal Microphone (Conexant

Format: AAC

Channels: Stereo

Sample Rate: 44100 Hz

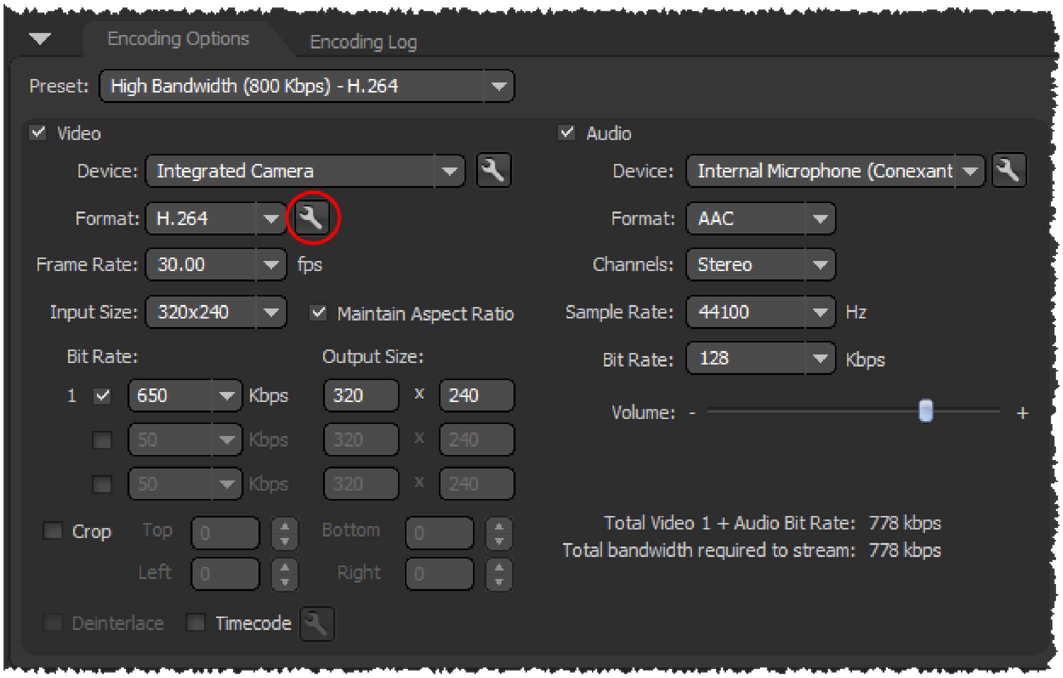
Bit Rate: 128 Kbps

Volume: - +

Total Video 1 + Audio Bit Rate: 778 kbps

Total bandwidth required to stream: 778 kbps

10. In the **Video** section of the **Encoding Options** tab, click the wrench icon to the right of the **Format** list to open the **Advanced Encoder Settings** dialog box.



Encoding Options | Encoding Log

Preset: High Bandwidth (800 Kbps) - H.264

☑ Video

Device: Integrated Camera

Format: H.264

Frame Rate: 30.00 fps

Input Size: 320x240 ☑ Maintain Aspect Ratio

Bit Rate: / Output Size:

1 ☑ 650 Kbps 320 x 240

□ 50 Kbps 320 x 240

□ 50 Kbps 320 x 240

□ Crop Top 0 Bottom 0

Left 0 Right 0

□ Deinterlace □ Timecode

☑ Audio

Device: Internal Microphone (Conexant

Format: AAC

Channels: Stereo

Sample Rate: 44100 Hz

Bit Rate: 128 Kbps

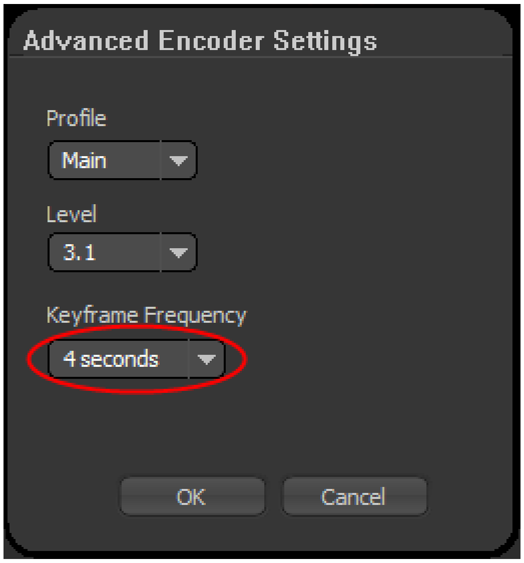
Volume: - +

Total Video 1 + Audio Bit Rate: 778 kbps

Total bandwidth required to stream: 778 kbps

11. In the **Advanced Encoder Settings** dialog box, for **Keyframe Frequency**, select **4 Seconds**.

    You can also use a multiple of the value of the <FragmentDuration> element in the `applications/livepkgr/events/_definst_/liveevent/Event.xml` file. The default value of <FragmentDuration> is 4000 milliseconds (4 seconds).



Advanced Encoder Settings

Profile: Main

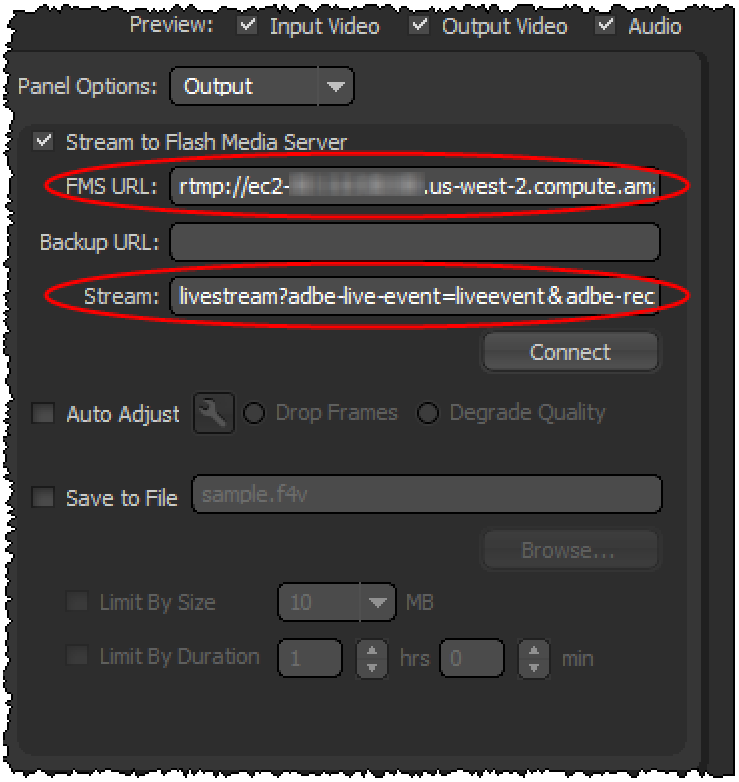
Level: 3.1

Keyframe Frequency: 4 seconds

12. Click **OK** to save the setting and return to the main page. The selection for the **Preset** list changes to **Custom**.
13. Open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation/.
14. Check the checkbox for the stack that you created for live streaming.
15. In the bottom pane, click the **Outputs** tab.
16. Copy the value of the **AMSURL** key, for example, *rtmp://ec2-00-11-22-33.us-west-1.compute.amazonaws.com/livepkgr*.
17. In Flash Media Live Encoder, in the Stream to Flash Media Server section, in the **FMS URL** setting, paste the value of the AMSURL key that you copied from the AWS CloudFormation console.
18. In the AWS CloudFormation console, copy the value of the **Stream** key, for example, *livestream?adbe-live-event=liveevent*.
19. In Flash Media Live Encoder, in the **Stream** setting, paste the value of the **Stream** key that you copied from the AWS CloudFormation console.

    Note

    If you anticipate having to stop and restart the live stream, enter the following value in the **Stream** field instead:
    `livestream?adbe-live-event=liveevent&adbe-record-mode=record`
    If you publish a live stream in record mode (`adbe-record-mode=record`), and then stop the stream and restart it, Adobe Media Server will delete the previous stream and start a new stream instead of appending to the previous stream when you restart. However, if you don't use record mode and you stop the live stream, you have to reconfigure live streaming before you can restart the stream.



Preview: ☑ Input Video ☑ Output Video ☑ Audio

Panel Options: Output

☑ Stream to Flash Media Server

FMS URL: rtmp://ec2-[redacted].us-west-2.compute.am

Backup URL:

Stream: livestream?adbe-live-event=liveevent&adbe-rec

□ Auto Adjust ○ Drop Frames ○ Degrade Quality

□ Save to File sample.f4v

□ Limit By Size 10 MB

□ Limit By Duration 1 hrs 0 min

20. Uncheck **Save to File**.
21. Click **Connect** to connect to your Adobe Media Server instance.
22. Click **Start** to start encoding and publishing your live stream to the livepkgr application on your Adobe Media Server instance.

Next: Embedding Strobe Media Playback for an Amazon CloudFront Live HTTP Stream in a Web Application (p. 211)

Embedding Strobe Media Playback for an Amazon CloudFront Live HTTP Stream in a Web Application

Follow one of these procedures to get the embed code that you will include in your web page for the live stream:

- To embed Strobe Media Playback for your HTTP stream (p. 211)
- To play your live HLS stream on an Apple device by using CloudFront (p. 212)

To embed Strobe Media Playback for your HTTP stream

1. Download the latest version of Open Source Media Framework (OSMF), which contains Strobe Media Playback. OSMF is available at http://sourceforge.net/projects/osmf.adobe/files/.
2. Unzip the file that you downloaded in step 1.
3. In the location where you unzipped the downloaded file, find StrobeMediaPlayback.swf, and copy it to a location, such as an Amazon S3 bucket, that is accessible to your live-streaming users.
4. Confirm that your users have the permissions necessary to access StrobeMediaPlayback.swf.

5. Change permissions in the crossdomain.xml file to allow users to view the live stream using Strobe Media Playback. For more information, see How do I update crossdomain.xml for a Flash-based stream hosted on my own domain? (p. 215)
6. In the location where you unzipped the downloaded file, find setup.html and open it in a web browser.
7. On the **Change Your Flash Vars** page, in the **Embed Parameters** section, in the **Source** field, enter the full URL for StrobeMediaPlayback.swf. This is the file that you copied in step 3. For example:

   ```
   https://myawsbucket.s3.amazonaws.com/LiveStreaming/StrobeMediaPlayback.swf
   ```
8. Open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation/.
9. Select the stack for live streaming.
10. In the bottom pane, click the **Outputs** tab.
11. Copy the value of the **LiveHDSManifest** key, as shown in the following example:

    ```
    http://d123.cloudfront.net/hds-live/livep-
    kgr/_definst_/liveevent/livestream.f4m
    ```
12. Back on the Change Your Flash Vars page, in the **Flash Vars** section, in the **src** field, paste the value that you copied in step 11.
13. At the bottom of the Change Your Flash Vars page, click **Preview and Update**.
14. Play the video to ensure that you're satisfied with the current settings, and update the settings as needed.
15. If you change any settings, click **Preview and Update** again.
16. To embed Strobe Media Playback in a web page, copy the contents of the **Preview Code** text box, and paste it into the HTML code for your website.

To play your live HLS stream on an Apple device by using CloudFront

1. Open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation/.
2. Select the stack for live streaming.
3. In the bottom pane, click the **Outputs** tab.
4. Copy the value of the **LiveHLSManifest** key, as shown in the following example:

   ```
   http://d123.cloudfront.net/hls-live/livep-
   kgr/_definst_/liveevent/livestream.m3u8.
   ```
5. Navigate to this URL using an iOS device to verify that HLS streaming is working correctly.

For information about where to use the URL to serve various iOS devices, QuickTime, and Safari, see HTTP Live Streaming Overview in the iOS Developer Library.

For more information about publishing and playing live streams over HTTP, see URLs for publishing and playing live streams over HTTP in the *Adobe Media Server 5.0.1 Developer's Guide*.

Next: Deleting an AWS CloudFormation Stack and an Amazon EBS Volume for Live Streaming (p. 212)

Deleting an AWS CloudFormation Stack and an Amazon EBS Volume for Live Streaming

When your live event is over, delete the stack that you created for live streaming. This deletes most of the AWS resources that were created for your live-streaming event, and stops most of the AWS charges for the resources. In addition, delete the Amazon EBS volume that is created by AWS CloudFormation but is not deleted when you delete the stack. This stops the rest of the AWS charges for the resources.

To delete an AWS CloudFormation stack and an Amazon EBS volume for live streaming

1. Sign in to the AWS Management Console and open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation/.
2. Find the AWS CloudFormation stack that you created for live streaming, and make note of the creation time. This will help you identify the Amazon EBS volume that you'll delete later in this procedure.
3. Select the stack, and click **Delete Stack**.
4. Click **Yes, Delete** to confirm.
5. To track the progress of the stack deletion, select the stack, and click the **Events** tab in the bottom frame.
6. Navigate to the Amazon EC2 console.
7. In the navigation pane, click **Volumes**.
8. Select the volume that the AWS CloudFormation stack created, click **Actions**, and click **Delete Volume**.

   If you have multiple Amazon EBS volumes, use the date and time that you made note of in step 2 to locate the volume that the AWS CloudFormation stack created.
9. If you don't plan to use live streaming again soon, you can cancel your subscription to Adobe Media Server on AWS Marketplace. To cancel the subscription, go to your subscriptions page on AWS Marketplace, find the row for Adobe Media Server, click **Cancel Subscription**, and follow the on-screen prompts.

Frequently Asked Questions

- How can I use Secure Shell (SSH) to connect to my Amazon EC2 instance that is running Adobe Media Server 5.0? (p. 213)
- How do I update crossdomain.xml for a Flash-based stream hosted on my own domain? (p. 215)
- What is the price for live HTTP streaming using CloudFront and Adobe Media Server 5.0? (p. 216)
- How can I create a CNAME alias for my Amazon EC2 instance or for my CloudFront distribution? (p. 216)
- How can I connect to the Adobe Media Server Administration Console? (p. 216)
- Can I stream my live event both to Apple devices and to Flash Player–compatible devices? (p. 217)
- Does Adobe Media Server 5.0 support HTML5? (p. 217)
- Does Adobe Media Server have logging? (p. 218)
- How can I enable authentication on Adobe Media Server? (p. 218)
- What are the default cache-control settings on HDS- and HLS-related files? (p. 218)
- What is the difference between HLS and HDS? (p. 218)
- How do I troubleshoot my Amazon EC2 instance if streaming doesn't start? (p. 219)
- Where can I find the documentation for live streaming using Adobe Flash Media Server 4.5? (p. 219)

How can I use Secure Shell (SSH) to connect to my Amazon EC2 instance that is running Adobe Media Server 5.0?

> **Note**
> By default, the SSH port for the Amazon EC2 instance (port 22) is disabled for security reasons. The following procedure explains how to enable the SSH port and how to use SSH to connect to your Amazon EC2 instance.

To enable access to port 22 on your Amazon EC2 instance that is running Adobe Media Server 5.0

1. Get the name of the Amazon EC2 security group that is associated with your Amazon EC2 instance:

   a. Sign in to the AWS Management Console and open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation/.

   b. In the Region list, select the region in which you created your Amazon EC2 instance.

   c. Click the row for your AWS CloudFormation stack.

   d. In the bottom pane, click the **Resources** tab.

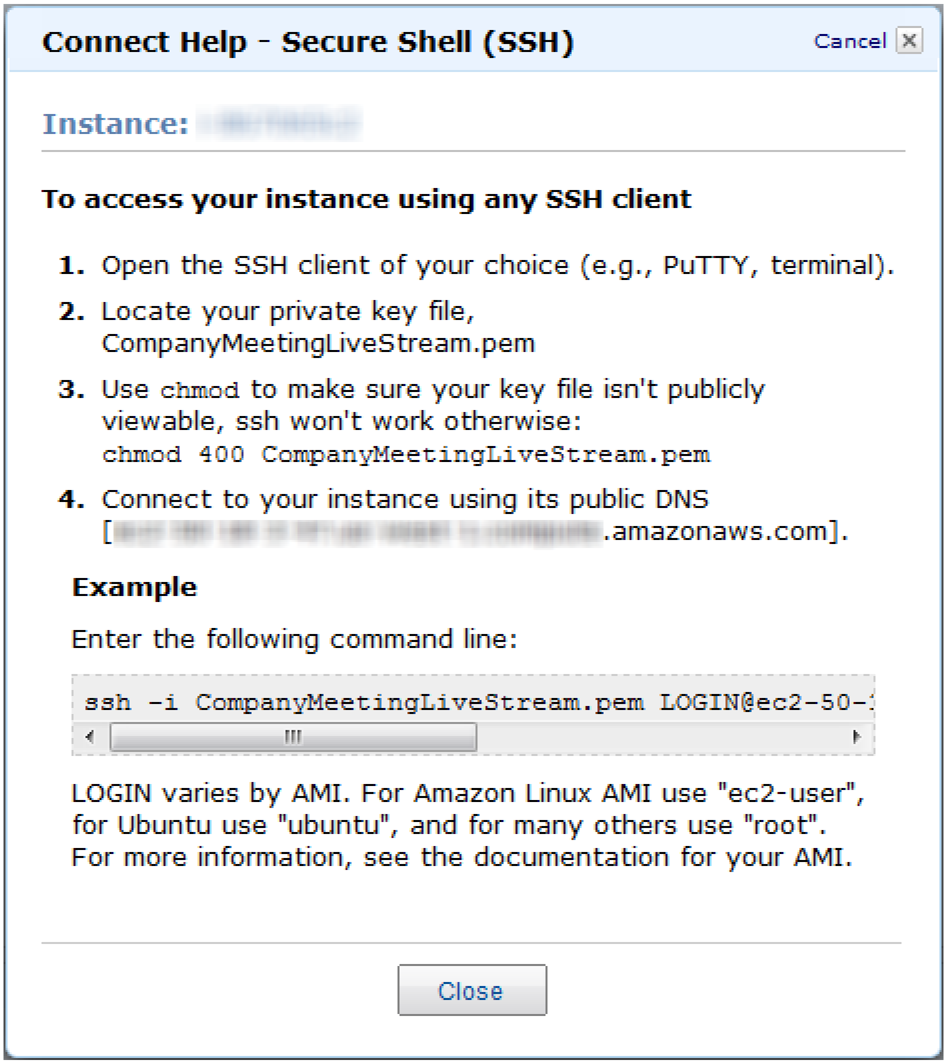
   e. In the left column of the **Stack Resources** table, find the row for which the value is `AMSOrigin-ServerSecurityGroup`.

   f. For that row, write down the value of the **Physical ID** column.

2. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
3. In the navigation pane, click **Security Groups**.
4. On the Security Groups page, select the row in which the **Name** column matches the physical ID that you got in Step 1f.
5. In the bottom pane, click the **Inbound** tab.
6. For **Create a new rule**, select **SSH**.
7. Click **Add Rule**.
8. Click **Apply Rule Changes**.

To use SSH to connect to your Amazon EC2 instance that is running Adobe Media Server 5.0

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Instances**.
3. Right-click the correct instance, and click **Connect** to view instructions on how to use SSH to connect to your Amazon EC2 instance.



Connect Help - Secure Shell (SSH)

Instance: [redacted]

To access your instance using any SSH client

1. Open the SSH client of your choice (e.g., PuTTY, terminal).
2. Locate your private key file, CompanyMeetingLiveStream.pem
3. Use `chmod` to make sure your key file isn't publicly viewable, ssh won't work otherwise:
   `chmod 400 CompanyMeetingLiveStream.pem`
4. Connect to your instance using its public DNS [[redacted].amazonaws.com].

Example

Enter the following command line:

```
ssh -i CompanyMeetingLiveStream.pem LOGIN@ec2-50-
```

LOGIN varies by AMI. For Amazon Linux AMI use "ec2-user", for Ubuntu use "ubuntu", and for many others use "root". For more information, see the documentation for your AMI.

How do I update crossdomain.xml for a Flash-based stream hosted on my own domain?

You can change permissions in `crossdomain.xml` either before or after you create the AWS CloudFormation stack:

- If you have not created your AWS CloudFormation stack, download the AWS CloudFormation template for Live Streaming using Amazon CloudFront and Adobe Media Server 5.0 at https://s3.amazonaws.com/cloudfront-live/live-http-streaming-ams-5-0-1-using-cloudfront.txt. In the template, edit the `UserData` section, which contains the `crossdomain.xml` settings, and save the updated template on your local computer. Then create your AWS CloudFormation stack using the updated template.
- If you have already created your AWS CloudFormation stack, log in to Adobe Media Server running on your Amazon EC2 instance, and change permissions in the cross-domain policy file, `/mnt/webroot/crossdomain.xml`.

For more information about editing the `crossdomain.xml` file, see Adobe Cross Domain Policy File Specification.

What is the price for live HTTP streaming using CloudFront and Adobe Media Server 5.0?

In addition to the $5.00 monthly subscription fee for Adobe Media Server on Amazon EC2, you pay only for the AWS resources you consume:

- For pricing information about Adobe Media Server running on Amazon EC2, see Adobe Media Server 5 on Amazon Web Services / Pricing.
- For pricing information about CloudFront, see Amazon CloudFront Pricing.

There is no charge for using AWS CloudFormation.

How can I create a CNAME alias for my Amazon EC2 instance or for my CloudFront distribution?

Your Amazon EC2 instance running Adobe Media Server 5.0 comes with an internal and an external DNS name. Amazon EC2 does not provide access to modify these DNS settings. If you want to map an existing domain name to your Amazon EC2 instance running Adobe Media Server, use a DNS service provider such as Amazon Route 53. When using your own domain name, we recommend that you map to the instance's external DNS name using a CNAME, not by using an A record that points to the instance's IP address.

To map your own domain name to your CloudFront distribution, see Using Alternate Domain Names (CNAMEs) (p. 29).

How can I connect to the Adobe Media Server Administration Console?

To connect to the Adobe Media Server Administration Console

1. Sign in to the AWS Management Console and open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation/.
2. Select the stack for live streaming.
3. In the bottom pane, click the **Outputs** tab.
4. Copy the value of the **AMSAdminConsoleServerAddress** key.
5. Click the value of the **AMSServerAdminConsole** key, for example, **http://ec2-00-11-22-33.us-west-1.compute.amazonaws.com/ams_adminConsole.htm**.
6. On the login page for the Adobe Media Server Administration Console, in **Server Address**, paste the **AMSAdminConsoleServerAddress** key that you copied in Step 4.
7. In the **Username** and **Password** fields, enter the values that you specified in Creating an AWS CloudFormation Stack for Live Streaming (p. 206).
8. Click **Login**.

For information about using the Adobe Media Server 5.0 Administration Console, see the Adobe Media Server documentation.

> **Note**
> Adobe recommends that you block all external access to port 1111 so that access to the Administration Console is restricted only to clients that are within your firewall. As an alternative, you can restrict access to the server by using domain-based restrictions. For more information, see Limit access to Adobe Media Administration Server in the Adobe documentation.

To disable or restrict access to port 1111 on your Adobe Media Server

1. Get the name of the Amazon EC2 security group that is associated with your Amazon EC2 instance:

   a. Sign in to the AWS Management Console and open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation/.
   b. For **Region**, click the name of the region in which you created your Amazon EC2 instance.
   c. Select the row for your AWS CloudFormation stack.
   d. In the bottom pane, click the **Resources** tab.
   e. In the **Stack Resources** table, in the `AMSOriginServerSecurityGroup` row, write down the value of the **Physical ID** column.

2. Display the Amazon EC2 console.
3. In the navigation pane, click **Security Groups**.
4. In the **Security Groups** pane, select the security group that AWS CloudFormation created for your Amazon EC2 instance. The name is the value that you wrote down in Step 1e.
5. In the bottom pane, click the **Inbound** tab.
6. To completely disable access to the Adobe Media Server Administration console:

   a. In the **TCP Port (Service)** column, find **1111**.
   b. In the **Action** column for that row, click **Delete**.
   c. Click **Apply Rule Changes**.

7. To restrict access to selected IP addresses:

   a. In the **TCP Port (Service)** column, find **1111**, and click **Delete**.
   b. For **Create a new rule**, accept the default value, **Custom TCP rule**.
   c. For **Port range**, enter **1111**.
   d. For **Source**, enter an IP address or range, or enter the name of another security group. For more information, click **Help**.
   e. Click **Add Rule**.
   f. To create additional rules, repeat Steps b through e.
   g. Click **Apply Rule Changes**.

Can I stream my live event both to Apple devices and to Flash Player–compatible devices?

Yes, Adobe Media Server 5.0 enables the delivery of live streams to both Flash-based and iOS devices at the same time. You can stream to the Safari browser using an HTML5 player or an Objective C ("native") application. You can also use Adobe AIR for iOS to develop a rich video experience on iOS.

Does Adobe Media Server 5.0 support HTML5?

Yes. Adobe Media Server can deliver content to HTML5 on Apple iOS devices using the HLS streaming format. For other browsers supporting HTML5, you can use Adobe Media Server to deliver progressively.

Does Adobe Media Server have logging?

Yes. W3C-compliant ASCII logs, a real-time usage monitor, and a complete API for server and stream events help to ensure that you have all the tools you need to track and generate reports on your audience's content use. For more information about monitoring and managing log files in Adobe Media Server 5.0, see Monitoring and Managing Log Files in the Adobe documentation.

How can I enable authentication on Adobe Media Server?

You can restrict access to RTMP port 1935 (for both TCP and UDP) in the security group created by AWS CloudFormation for your Adobe Media Server Amazon EC2 instance. Just create new TCP and UDP rules for port 1935 and then delete the existing TCP and UDP rules for port 1935, which allow access to all IP addresses.

For a quick overview of how to add a rule to a security group, see How can I connect to the Adobe Media Server Administration Console? (p. 216). For more information about Amazon EC2 security groups, see Amazon EC2 Security Groups in the *Amazon Elastic Compute Cloud User Guide for Linux*.

What are the default cache-control settings on HDS- and HLS-related files?

The default cache control headers on HDS- and HLS-related files are set to the following values:

File Type	`Cache-Control` Setting (Seconds)
.bootstrap	2
HDS Fragment	60
.f4m	2
.m3u8	2
.ts	60

The CloudFront edge cache servers honor these cache control headers. You can change the default settings by changing the values of the `HttpStreamingF4MMaxAge`, `HttpStreamingBootstrapMaxAge`, and `HttpStreamingFragMaxAge` parameters on the server. For more information, see HTTP streaming configuration file reference in the Adobe documentation.

What is the difference between HLS and HDS?

HLS is a file container format optimized for Apple devices. The container supports H.264/AAC-encoded video and audio, and is based on MPEG-2 transport stream (TS). All video delivered to iOS (including AIR for IOS) must use this format.

HDS is a file container format optimized for applications that run in Flash Player. The container also supports H.264/AAC-encoded video and audio and is based on MPEG-4 TS. HDS is not supported on AIR for iOS.

How do I troubleshoot my Amazon EC2 instance if streaming doesn't start?

If you performed the procedure To verify that Adobe Media Server is running (p. 207) and streaming still hasn't started, perform the following procedure to confirm that the Amazon EC2 instance is functioning correctly.

To troubleshoot your Amazon EC2 instance running Adobe Media Server 5.0

1. In the AWS CloudFormation console, in the top pane, select the stack.
2. In the bottom pane, click the **Resources** tab.
3. For the **AMSOriginServer** row, write down the value of the **Physical ID** column.
4. Go to the Amazon EC2 console.
5. In the **Region** list, select the region in which you created the AWS CloudFormation stack.
6. In the navigation pane, click **Instances**.
7. In the **Instance** column, find the value that you wrote down in Step c.
8. Select the corresponding row.
9. In the bottom pane, review the information on the **Status Checks** tab, and take the recommended actions.
10. Return to the procedure To verify that Adobe Media Server is running (p. 207), and repeat Steps 2 through 5.

Where can I find the documentation for live streaming using Adobe Flash Media Server 4.5?

For the documentation for live streaming using Adobe Flash Media Server 4.5, see "Live Streaming Using CloudFront and Adobe Flash Media Server 4.5" in the "CloudFront Tutorials" chapter of the Amazon CloudFront Developer Guide for CloudFront API version 2012-07-01.

Additional Documentation

Adobe Documentation

- Using Adobe Media Server on Amazon Web Services
- Adobe Cross Domain Policy File Specification
- Flash Media Live Encoder
- Flash Media Live Encoder FAQ
- Video Encoding and Transcoding Recommendations for HTTP Dynamic Streaming on the Adobe Media Server Platform
- Adobe Media Server Technical Overview

Amazon Web Services Documentation

- Amazon Elastic Compute Cloud documentation
- AWS CloudFormation documentation

Live Smooth Streaming Using Amazon CloudFront and IIS Media Services 4.1

Topics

- Overview of Live Smooth Streaming with Amazon Web Services (p. 220)
- Creating an Amazon Web Services Account (p. 221)
- Creating an Amazon EC2 Key Pair (p. 221)
- Creating an AWS CloudFormation Stack for Live Smooth Streaming (p. 222)
- Verifying that Your Amazon EC2 Windows Server Instance Is Running (p. 225)
- Getting Your Windows Password (p. 225)
- Encoding Your Live Stream (p. 226)
- Viewing Your Live Smooth Stream (p. 231)
- Deleting Your AWS CloudFormation Live Smooth Streaming Stack (p. 231)
- Frequently Asked Questions (p. 232)
- Additional Documentation (p. 234)

Overview of Live Smooth Streaming with Amazon Web Services

Smooth Streaming is the Microsoft implementation of adaptive streaming technology, which is a form of web-based media content delivery that uses standard HTTP. An extension of IIS Media Services, Smooth Streaming enables adaptive streaming of live events to Smooth Streaming clients such as Microsoft Silverlight. When you configure Smooth Streaming to use CloudFront, you benefit from the scale of CloudFront's global HTTP network and from latency-based routing of viewers to edge nodes on the network. To learn more about CloudFront, go to the CloudFront product page.

Smooth Streaming content is delivered to clients as a series of MPEG-4 (MP4) fragments that can be cached at the CloudFront edge servers. Smooth Streaming–compatible clients use special heuristics to dynamically monitor current network and local PC conditions, and seamlessly switch the video quality of the Smooth Streaming presentation that the clients receive. As clients play the fragments, network conditions may change (for example, bandwidth may decrease) or video processing may be affected by other applications that are running on the client. Clients can immediately request that the next fragment come from a stream that is encoded at a different bit rate to accommodate the changing conditions. This enables clients to play the media without stuttering, buffering, or freezing. As a result, users experience the highest-quality playback available without interruptions in the stream.

To encode a live broadcast to Smooth Streaming format, you use Microsoft Expression Encoder 4 Pro. To serve the encoded Smooth Stream, you can then use an Amazon EC2 Amazon Machine Image (AMI) that is running Windows IIS Media Services. CloudFront caches the live video and audio content, and viewers connect to the CloudFront edge servers to play the stream using a Smooth Streaming-compatible client such as Microsoft Silverlight. This tutorial walks you through the entire setup process.

> **Note**
> Microsoft Expression Encoder 4 Pro with Service Pack 2 is not included in the Amazon EC2 Amazon Machine Image (AMI) that is running Windows IIS Media Services, and it is not a free download. For information about features and pricing, go to the Expression Encoder 4 Pro page on the Microsoft Store website. You can also use a third-party encoding tool to encode your video for Live Smooth Streaming. For a list of Microsoft partners that provide encoding software, see the Partners tab on the IIS Media Services page on the Microsoft website.

> **Note**
> This tutorial provides an overview of how to integrate CloudFront with Microsoft Live Smooth Streaming running on an Amazon EC2 instance. For more information about how to manage and secure your Amazon EC2 instance, refer to the Amazon EC2 documentation. For more information about Microsoft Live Smooth Streaming options not covered in this tutorial, see Microsoft Documentation (p. 234).

To set up Live Smooth Streaming with Amazon Web Services (AWS), review the system requirements for IIS Smooth Streaming in the *Smooth Streaming Deployment Guide*. Then perform the procedures in the following sections:

1. Creating an Amazon Web Services Account (p. 221)
2. Creating an Amazon EC2 Key Pair (p. 221)
3. Creating an AWS CloudFormation Stack for Live Smooth Streaming (p. 222)
4. Verifying that Your Amazon EC2 Windows Server Instance Is Running (p. 225)
5. Getting Your Windows Password (p. 225)
6. Encoding Your Live Stream (p. 226)
7. Viewing Your Live Smooth Stream (p. 231)
8. Deleting Your AWS CloudFormation Live Smooth Streaming Stack (p. 231)

For frequently asked questions, see Frequently Asked Questions (p. 232).

For links to additional Microsoft and AWS documentation, see Additional Documentation (p. 234).

Creating an Amazon Web Services Account

If you already have an AWS account, skip to Creating an Amazon EC2 Key Pair (p. 221). If you don't already have an AWS account, use the following procedure to create one.

> **Note**
> When you create an account, AWS automatically signs up the account for all services. You are charged only for the services you use.

To create an AWS account

1. Go to http://aws.amazon.com, and click **Create an AWS Account**.
2. Follow the on-screen instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.

Next: Creating an Amazon EC2 Key Pair (p. 221)

Creating an Amazon EC2 Key Pair

If you already have an Amazon EC2 key pair in the Amazon EC2 region in which you want to configure Live Smooth Streaming, skip to Creating an AWS CloudFormation Stack for Live Smooth Streaming (p. 222). If you don't have a key pair in that region, perform the following procedure.

A key pair is a security credential similar to a password. You specify a key pair when you create an AWS CloudFormation stack for live streaming, later in this process. After live streaming is configured, you use the key pair to retrieve the password for your Amazon EC2 Windows Server instance.

To create an Amazon EC2 key pair

1. Sign in to the AWS Management Console and open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the Region list, click the region in which you want to create the key pair.

   You must create the key pair in the same region where you will create your AWS CloudFormation stack for live streaming later in this process. We recommend that you create the key pair and the stack for live streaming in the region that is closest to the location of your live event.
3. In the Navigation pane, click **Key Pairs**.
4. In the Key Pairs pane, click **Create Key Pair**.
5. In the Create Key Pair dialog box, enter a name for the key pair, and make note of the name. You'll enter this value when you create an AWS CloudFormation live-streaming stack, later in the process of setting up live streaming.
6. Click **Create**, and the Opening <key_pair_name>.pem dialog box appears.
7. Save the .pem file to a safe place on your computer.
8. Click **Close** to close the Create Key Pair dialog box.

Next: Creating an AWS CloudFormation Stack for Live Smooth Streaming (p. 222)

Creating an AWS CloudFormation Stack for Live Smooth Streaming

The following procedure uses an AWS CloudFormation template to create a stack that launches the AWS resources required for Live Smooth Streaming, including an Amazon EC2 instance.

> **Important**
> You incur hourly charges for an Amazon EC2 instance beginning when you create the AWS CloudFormation stack that deploys the Amazon EC2 instance. Charges continue to accrue until you delete the AWS CloudFormation stack regardless of whether you use the Amazon EC2 instance to stream live video. For more information, see Pricing on the Amazon Elastic Compute Cloud (Amazon EC2) detail page. When your live event is over, delete the stack that you created for Live Smooth Streaming. This deletes the AWS resources that were created for your live-streaming event, and stops the AWS charges for the resources. For more information, see Deleting Your AWS CloudFormation Live Smooth Streaming Stack (p. 231).

To create an AWS CloudFormation stack for live streaming

1. In the following list, click the Amazon EC2 Region where you want to create the stack. The Create Stack wizard starts, and a region-specific value is automatically entered in the **Provide a Template URL** field.

   US East (Virginia)

   US West (Oregon)

   US West (Northern California)

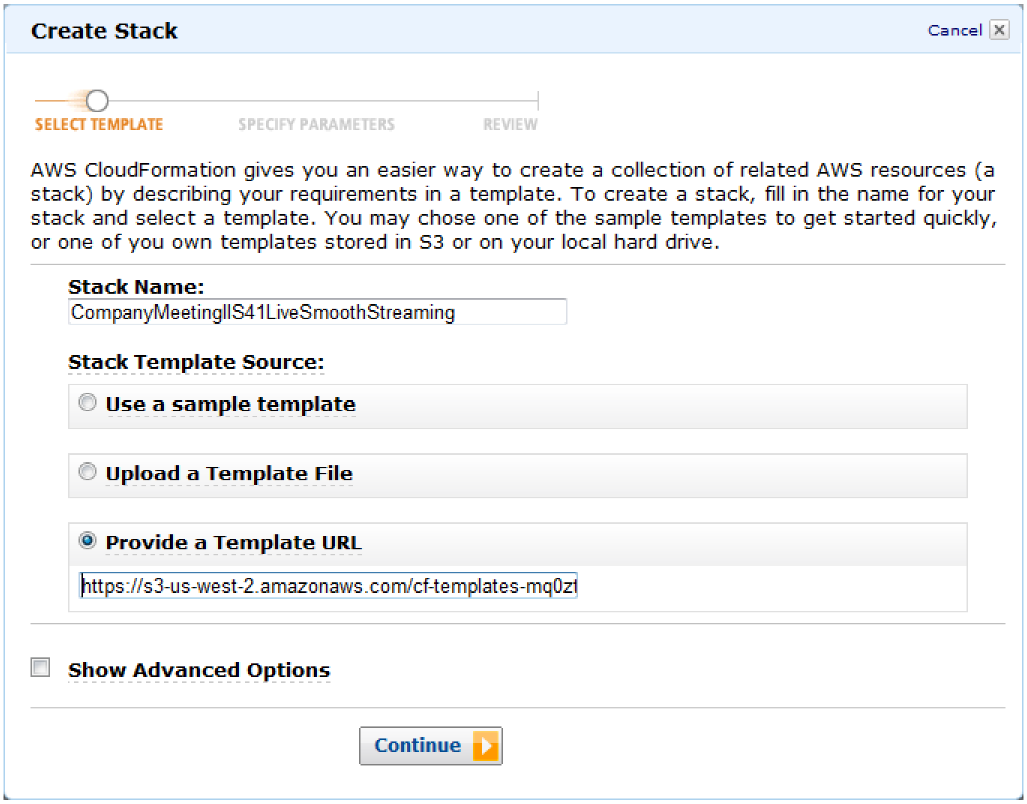
   EU (Ireland)

   Asia Pacific (Singapore)

   Asia Pacific (Tokyo)

   South America (Sao Paulo)

2. If you are not already signed in to the AWS Management Console, sign in when prompted.
3. *Optional:* In the Create Stack wizard, change the value of the **Stack Name** field. The stack name must not contain spaces, and it must be unique within your AWS account.



AWS CloudFormation gives you an easier way to create a collection of related AWS resources (a stack) by describing your requirements in a template. To create a stack, fill in the name for your stack and select a template. You may chose one of the sample templates to get started quickly, or one of you own templates stored in S3 or on your local hard drive.

Stack Name: CompanyMeetingIIS41LiveSmoothStreaming

Stack Template Source:

○ Use a sample template

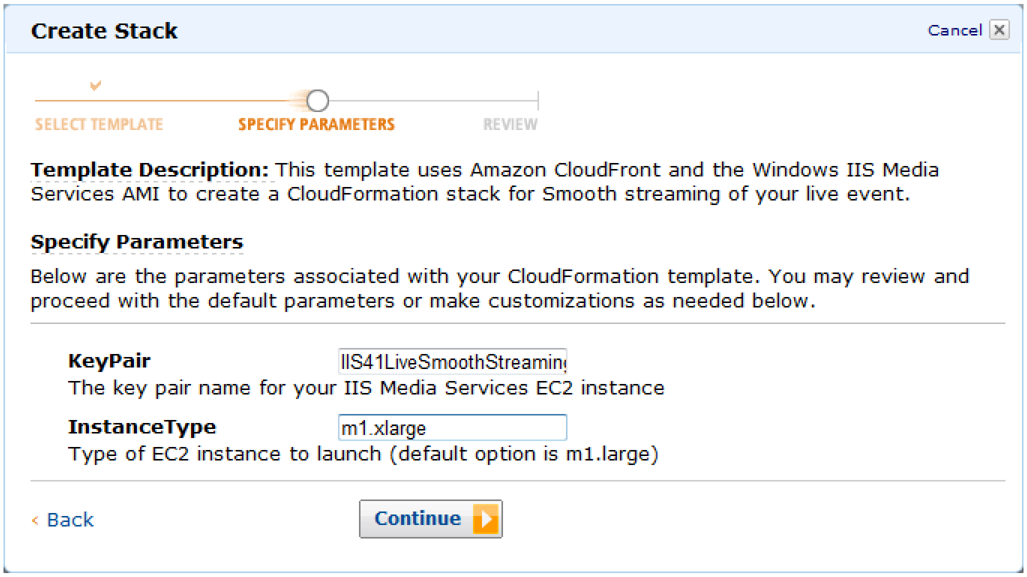
○ Upload a Template File

● Provide a Template URL

https://s3-us-west-2.amazonaws.com/cf-templates-mq0zt

□ Show Advanced Options

4. Do not change the **Stack Template Source** option or the value of **Provide a Template URL**.
5. *Optional:* To configure SNS notification, to specify how long you're willing to wait for the stack to be created, and to choose whether to roll back changes if stack creation fails, check the **Show Advanced Options** checkbox, and specify the applicable values.
6. Click **Continue**.



Create Stack Cancel ☒

Template Description: This template uses Amazon CloudFront and the Windows IIS Media Services AMI to create a CloudFormation stack for Smooth streaming of your live event.

Specify Parameters

Below are the parameters associated with your CloudFormation template. You may review and proceed with the default parameters or make customizations as needed below.

KeyPair IIS41LiveSmoothStreamin
The key pair name for your IIS Media Services EC2 instance

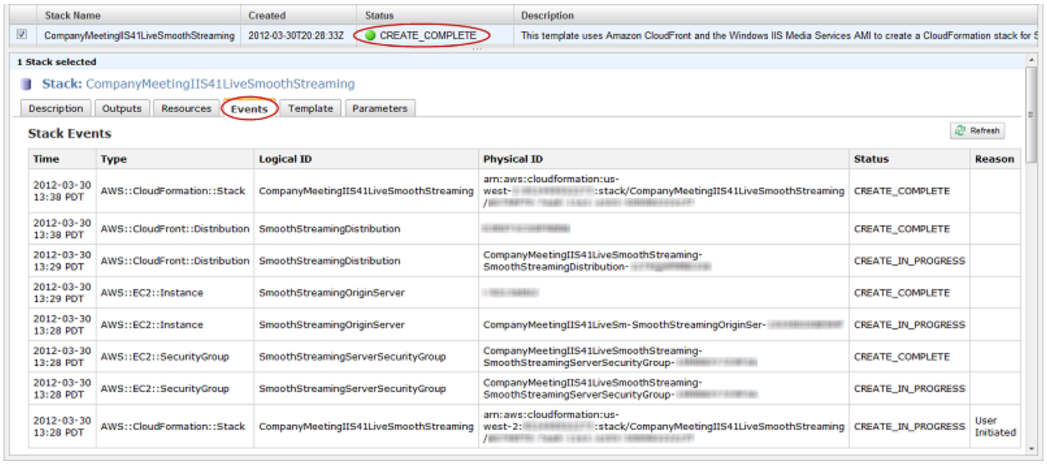
InstanceType m1.xlarge
Type of EC2 instance to launch (default option is m1.large)

7. On the Specify Parameters page, in the **KeyPair** field, enter the name of an Amazon EC2 key pair in the region in which you want to create the stack for live streaming. The key pair must be associated with the account that you're currently logged on with. If you created a key pair when you performed the procedure in Creating an Amazon EC2 Key Pair (p. 221), enter the name of that key pair.
8. In the **InstanceType** field, enter an instance type, and click **Continue**. The default value is *m1.xlarge*.

   The instance type determines the pricing for your Amazon EC2 instance that is running Windows Server. For more information about Amazon EC2 instance types for Windows, including pricing information, go to Amazon EC2 Running Microsoft Windows Server & SQL Server.
9. Review the settings for the stack. When you're satisfied with the settings, click **Create Stack**.

   Your stack may take several minutes to create. To track the progress of the stack creation, select the stack, and click the **Events** tab in the bottom frame. If AWS CloudFormation cannot create the stack, the Events tab lists error messages.

   When your stack is ready, in the top frame, the status for the stack changes to **CREATE_COMPLETE**.



	Stack Name	Created	Status	Description
☑	CompanyMeetingIIS41LiveSmoothStreaming	2012-03-30T20:28:33Z	● CREATE_COMPLETE	This template uses Amazon CloudFront and the Windows IIS Media Services AMI to create a CloudFormation stack for S

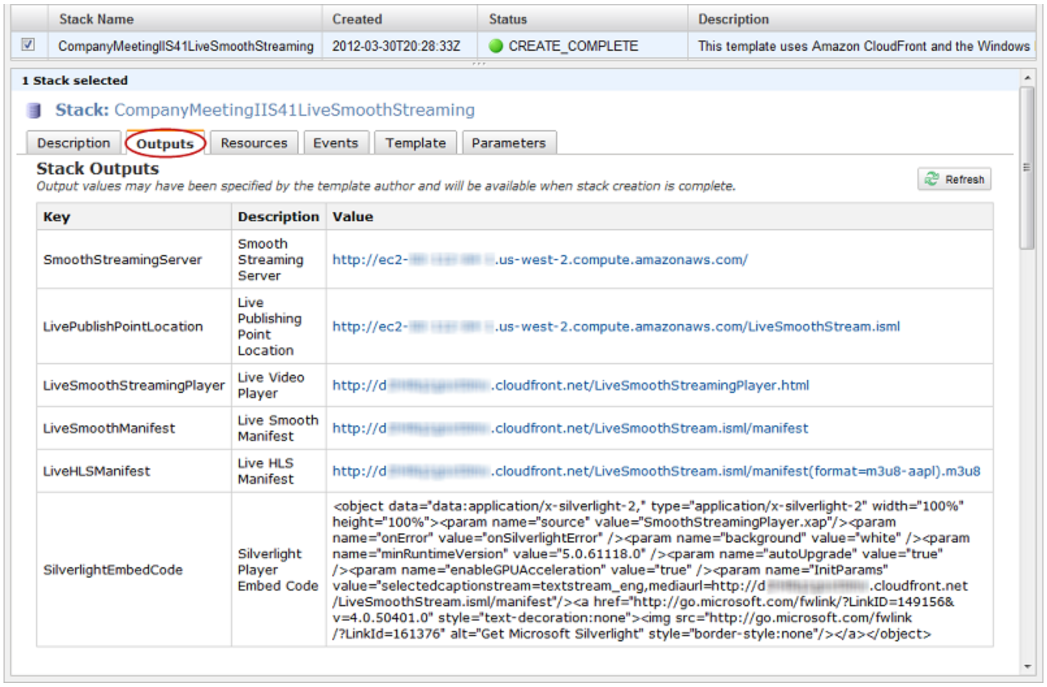
1 Stack selected

Stack: CompanyMeetingIIS41LiveSmoothStreaming

Stack Events

Time	Type	Logical ID	Physical ID	Status	Reason
2012-03-30 13:38 PDT	AWS::CloudFormation::Stack	CompanyMeetingIIS41LiveSmoothStreaming	arn:aws:cloudformation:us-west-[redacted]:stack/CompanyMeetingIIS41LiveSmoothStreaming /[redacted]	CREATE_COMPLETE	
2012-03-30 13:38 PDT	AWS::CloudFront::Distribution	SmoothStreamingDistribution	[redacted]	CREATE_COMPLETE	
2012-03-30 13:29 PDT	AWS::CloudFront::Distribution	SmoothStreamingDistribution	CompanyMeetingIIS41LiveSmoothStreaming-SmoothStreamingDistribution-[redacted]	CREATE_IN_PROGRESS	
2012-03-30 13:29 PDT	AWS::EC2::Instance	SmoothStreamingOriginServer	[redacted]	CREATE_COMPLETE	
2012-03-30 13:28 PDT	AWS::EC2::Instance	SmoothStreamingOriginServer	CompanyMeetingIIS41LiveSm-SmoothStreamingOriginSer-[redacted]	CREATE_IN_PROGRESS	
2012-03-30 13:28 PDT	AWS::EC2::SecurityGroup	SmoothStreamingServerSecurityGroup	CompanyMeetingIIS41LiveSmoothStreaming-SmoothStreamingServerSecurityGroup-[redacted]	CREATE_COMPLETE	
2012-03-30 13:28 PDT	AWS::EC2::SecurityGroup	SmoothStreamingServerSecurityGroup	CompanyMeetingIIS41LiveSmoothStreaming-SmoothStreamingServerSecurityGroup-[redacted]	CREATE_IN_PROGRESS	
2012-03-30 13:28 PDT	AWS::CloudFormation::Stack	CompanyMeetingIIS41LiveSmoothStreaming	arn:aws:cloudformation:us-west-2:[redacted]:stack/CompanyMeetingIIS41LiveSmoothStreaming /[redacted]	CREATE_IN_PROGRESS	User Initiated

When your stack is created, click the **Outputs** tab, which displays the stack creation outputs. You will use these values when you set up Microsoft Expression Encoder later in the process.



	Stack Name	Created	Status	Description
☑	CompanyMeetingIIS41LiveSmoothStreaming	2012-03-30T20:28:33Z	● CREATE_COMPLETE	This template uses Amazon CloudFront and the Windows

1 Stack selected

Stack: CompanyMeetingIIS41LiveSmoothStreaming

Stack Outputs

Output values may have been specified by the template author and will be available when stack creation is complete.

Key	Description	Value
SmoothStreamingServer	Smooth Streaming Server	http://ec2-[redacted].us-west-2.compute.amazonaws.com/
LivePublishPointLocation	Live Publishing Point Location	http://ec2-[redacted].us-west-2.compute.amazonaws.com/LiveSmoothStream.isml
LiveSmoothStreamingPlayer	Live Video Player	http://d[redacted].cloudfront.net/LiveSmoothStreamingPlayer.html
LiveSmoothManifest	Live Smooth Manifest	http://d[redacted].cloudfront.net/LiveSmoothStream.isml/manifest
LiveHLSManifest	Live HLS Manifest	http://d[redacted].cloudfront.net/LiveSmoothStream.isml/manifest(format=m3u8-aapl).m3u8
SilverlightEmbedCode	Silverlight Player Embed Code	<object data="data:application/x-silverlight-2," type="application/x-silverlight-2" width="100%" height="100%"><param name="source" value="SmoothStreamingPlayer.xap"/><param name="onError" value="onSilverlightError" /><param name="background" value="white" /><param name="minRuntimeVersion" value="5.0.61118.0" /><param name="autoUpgrade" value="true" /><param name="enableGPUAcceleration" value="true" /><param name="InitParams" value="selectedcaptionstream=textstream_eng,mediaurl=http://d[redacted].cloudfront.net /LiveSmoothStream.isml/manifest"/><a href="http://go.microsoft.com/fwlink/?LinkID=149156&v=4.0.50401.0" style="text-decoration:none"><img src="http://go.microsoft.com/fwlink /?LinkId=161376" alt="Get Microsoft Silverlight" style="border-style:none"/></a></object>

Next: Verifying that Your Amazon EC2 Windows Server Instance Is Running (p. 225)

Verifying that Your Amazon EC2 Windows Server Instance Is Running

After AWS CloudFormation creates the stack, perform the following procedure to verify that your Windows IIS Media Services webserver is running on the Amazon EC2 instance that you provisioned via AWS CloudFormation.

To verify that your Windows Server is running

1. Sign in to the AWS Management Console and open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation/.
2. In the top pane, select the stack that you created in Creating an AWS CloudFormation Stack for Live Smooth Streaming (p. 222).
3. In the bottom pane, click the **Outputs** tab.
4. Click on the value of the **SmoothStreamingServer** key, for example, **http://ec2-00-11-22-33.us-west-1.compute.amazonaws.com**.

   The Windows IIS Server banner screen appears, indicating that your Windows Server is running.

Next: Getting Your Windows Password (p. 225)

Getting Your Windows Password

To connect to your Amazon EC2 instance running Windows Server 2008 R2 and IIS Media Services, use the following procedure to retrieve the initial password for the Windows Server Administrator account. You only need to retrieve the password once for your Amazon EC2 instance. When you are finished with this procedure, you'll be able to work with your Amazon EC2 instance as you would any Windows Server computer.

For more information about connecting to an Amazon EC2 instance running Windows, go to Getting Started Guide AWS Computing Basics for Windows.

> **Important**
> Amazon EC2 can take as long as 30 minutes to retrieve your password from Windows Server.

To get the Windows password for your Amazon EC2 instance

1. Confirm that you can access the Amazon EC2 private key file (the .pem file) that you created in Creating an Amazon EC2 Key Pair (p. 221).
2. Sign in to the AWS Management Console and open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
3. In the **Region** list, click the region in which you created the Amazon EC2 instance for Live Smooth Streaming.
4. In the **Navigation** pane, click **Instances**.
5. In the My Instances pane, right-click the instance for which the value of the **Name** column is **LiveSmoothStreaming**, and click **Get Windows Password**.
6. On the Retrieve Default Windows Administrator Password page, click **Browse**, and browse to the location on your computer where you saved the .pem file.
7. Select the .pem file, and the contents of the file appear in the window.
8. Click **Decrypt Password**.
9. Write down the password. You'll need it to connect to the Amazon EC2 instance.
10. *Optional but recommended:* Log into the Windows Server instance that you just launched, and change the password for the default Windows Server account. The username is Administrator.

    You may also want to create another user account and add it to the Administrators group. Another administrator account is a safeguard in case you forget your administrator password or have a problem with the Administrator account.

> **Note**
> For information about how to update the Amazon EC2 Security Group settings for your Windows server so you can access the server using port 3389, see How can I enable access to the Windows server? (p. 233). For information about how to log on to the instance using the Administrator account, see How can I securely connect to my Amazon EC2 instance running Windows IIS Media Services? (p. 233).

Next: Encoding Your Live Stream (p. 226)

Encoding Your Live Stream

Use the procedure in this section to create a Live Broadcasting Project using Microsoft Expression Encoder 4 Pro SP2 and to publish your live stream to the Live Smooth Streaming publishing point on your Amazon EC2 instance running Windows Server and Windows IIS Media Services.

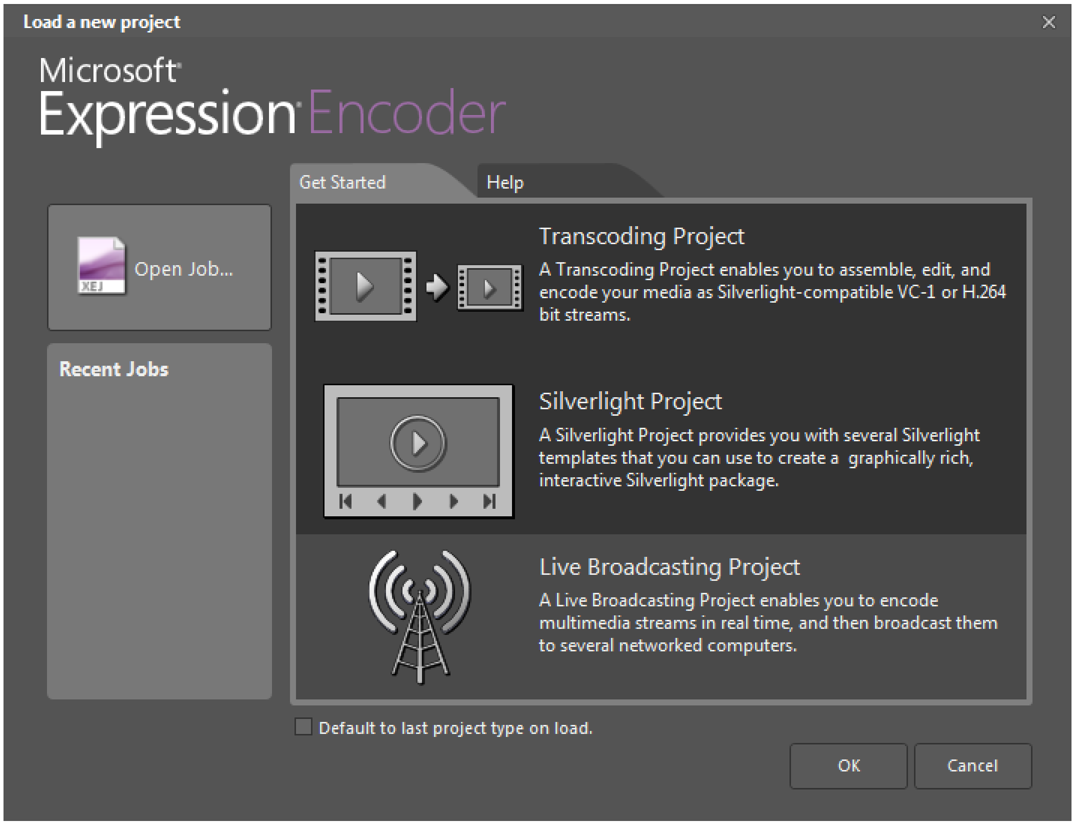
To learn more about live broadcasting using Microsoft Expression Encoder, go to Creating a Live Broadcasting Project on the Microsoft Expression website.

> **Note**
> Microsoft Expression Encoder 4 Pro with Service Pack 2 is not a free download. For more information about features and pricing, go to the Expression Encoder 4 Pro page on the Microsoft Store website.

You can also use a third-party encoding tool to encode your video for Live Smooth Streaming. For a list of Microsoft partners that provide encoding software, see the Partners tab on the IIS Media Services page on the Microsoft website.

To encode a live broadcast

1. Log on to the computer that you'll use to broadcast the live stream.
2. On the Windows Start menu, click **All Programs > Microsoft Expression > Microsoft Expression Encoder 4**.
3. In the **Load a new project** dialog box, click **Live Broadcasting Project**.



Transcoding Project

A Transcoding Project enables you to assemble, edit, and encode your media as Silverlight-compatible VC-1 or H.264 bit streams.

Silverlight Project

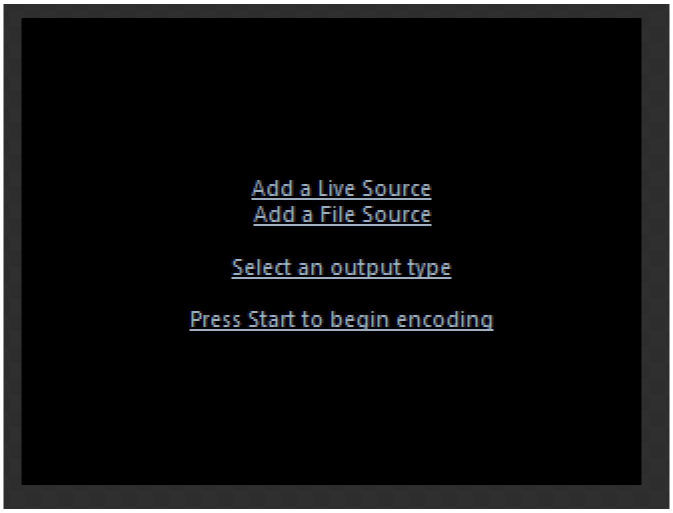
A Silverlight Project provides you with several Silverlight templates that you can use to create a graphically rich, interactive Silverlight package.

Live Broadcasting Project

A Live Broadcasting Project enables you to encode multimedia streams in real time, and then broadcast them to several networked computers.

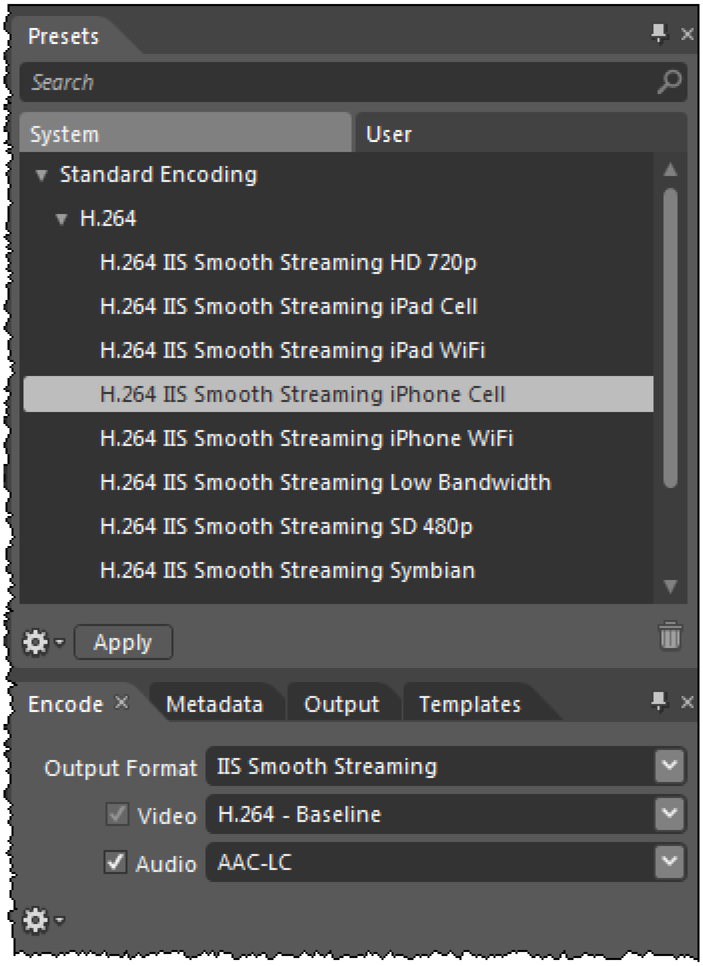
□ Default to last project type on load.

4. Click **Add a Live Source** to use for your live broadcast.



> **Note**
> You can connect multiple camera devices, such as USB webcams or FireWire (IEEE 1394) digital video cameras. Although you can connect multiple live sources, you can stream only one at a time. For more information about setting up sources for live broadcasts, go to Set Live Sources on the Microsoft Expression website.

5. On the **Presets** tab, choose the encoding preset that supports the bit rates and encoding requirements for your Live Smooth Streaming scenario. Choose an option that has **IIS Smooth Streaming** in the name.


Presets
Search
System
User
Standard Encoding
H.264
H.264 IIS Smooth Streaming HD 720p
H.264 IIS Smooth Streaming iPad Cell
H.264 IIS Smooth Streaming iPad WiFi
H.264 IIS Smooth Streaming iPhone Cell
H.264 IIS Smooth Streaming iPhone WiFi
H.264 IIS Smooth Streaming Low Bandwidth
H.264 IIS Smooth Streaming SD 480p
H.264 IIS Smooth Streaming Symbian
Apply
Encode
Metadata
Output
Templates

Output Format: IIS Smooth Streaming
☑ Video: H.264 - Baseline
☑ Audio: AAC-LC

When you click **Apply**, the **Output Format**, **Video**, and **Audio** settings on the **Encode** tab are automatically updated with the values in the encoding preset that you selected.

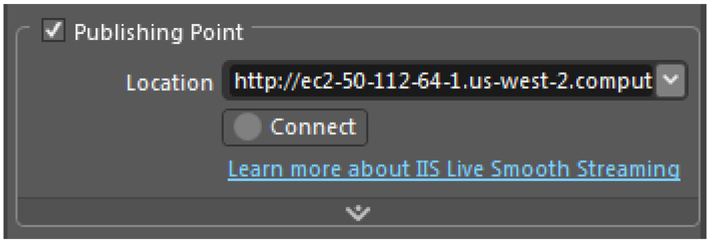
For more information about a preset, for example, the number of streams in the output and the codecs used, hover your mouse pointer over a preset name.

Note
Alternatively, you can specify custom settings on the **Encode** tab. For more information, go to the following topics on the Microsoft Expression website:

- Set Output formats
- Video settings
- Audio settings

6. In Microsoft Expression Encoder, click the **Output** tab.
7. On the **Output** tab, check the **Publishing Point** check box.
8. Sign in to the AWS Management Console and open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation/.
9. In the bottom pane of the AWS CloudFormation console, click the **Outputs** tab.
10. Copy the value of the **LivePublishPointLocation** key, for example, **http://ec2-00-11-22-33.us-west-1.compute.amazonaws.com/LiveSmoothStream.isml**.

11. In Microsoft Expression Encoder, paste the URL that you copied in the previous step into the **Location** field.



☑ Publishing Point

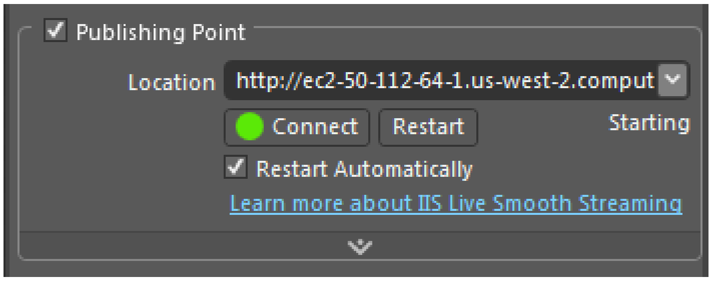
Location: http://ec2-50-112-64-1.us-west-2.comput

12. Click **Connect** to initiate a connection to the publishing point on your Windows server.
13. When you're prompted for your Publishing Point Administrator Password, enter the following values:

    - **User Name:** Administrator
    - **Password:** the Windows Server password that you retrieved in Getting Your Windows Password (p. 225).

    Then click **OK**.

    Note
    Windows authentication is configured for the default web site on your Windows server so you can connect to the live publishing point on the server from Microsoft Expression Encoder 4 Pro SP2. To learn more about Windows authentication, go to the IIS website. To learn about the authentication mechanisms available in IIS 7, go to the Microsoft website.
14. When a connection is successfully established, the publishing point state changes to **Starting**. In addition, a **Restart** button appears next to the **Connect** button, below the **Location** field in the Publishing Point section.



☑ Publishing Point

Location: http://ec2-50-112-64-1.us-west-2.comput

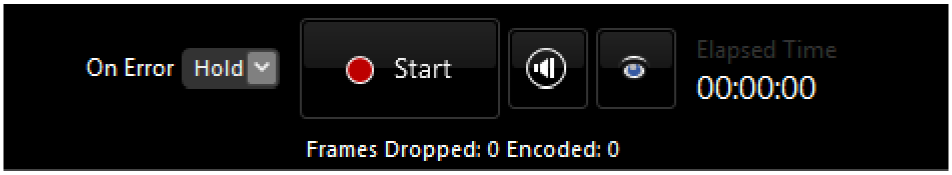
Starting

☑ Restart Automatically

Note
The Starting state means that the publishing point is ready to receive live streams. When the live source connects to the publishing point and begins pushing content to it, the Starting state changes to Started, meaning that the publishing point is receiving the live streams.

Note
Microsoft Expression Encoder 4 Pro with SP2 utilizes the REST APIs that Windows IIS Media Services 4.1 includes to help you manage live publishing points on the Windows server. For more information, go to the IIS blog.

15. Click **Start** to begin encoding and publishing your live broadcast to the publishing point on your Amazon EC2 instance that is running Windows Server and IIS Media Services.



As the broadcast runs, you can monitor statistics and connections data in the corresponding panels. For more information about how to monitor this data, see the following topics on the Microsoft Expression website:

- Using the Statistics panel
- Using the Connections panel

Next: Viewing Your Live Smooth Stream (p. 231)

Viewing Your Live Smooth Stream

Perform the following procedure to view your live smooth stream using CloudFront. You can also embed the Microsoft Silverlight player code in your own web page.

1. Sign in to the AWS Management Console and open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation/.
2. Select the stack for live streaming.
3. In the bottom pane of the AWS CloudFormation console, click the **Outputs** tab.
4. Click the value of the **LiveSmoothStreamingPlayer** key, for example, **http://d123.cloudfront.net/LiveSmoothStreamingPlayer.html**.
5. To embed the Silverlight player code into your web page, on the **Outputs** tab, copy the value of the **SilverlightEmbedCode** key.

   Note
   Microsoft recommends that viewers have the latest version of Microsoft Silverlight installed for the best playback experience.
6. To view your live stream on an Apple device such as an iPad or an iPhone, display the AWS CloudFormation console from a compatible Apple device, and click the value of the **LiveHLSManifest** key. The manifest URL looks like **http://d123.cloudfront.net/LiveSmoothStream.isml/manifest(format=m3u8-aapl).m3u8**.

   For information about where to use the URL to serve various iOS devices, QuickTime, and Safari, go to HTTP Live Streaming Overview in the iOS Developer Library.

Next: Deleting Your AWS CloudFormation Live Smooth Streaming Stack (p. 231)

Deleting Your AWS CloudFormation Live Smooth Streaming Stack

When your live event is over, delete the stack that you created for Live Smooth Streaming. This deletes the AWS resources that were created for your live event, and stops the AWS charges for those resources.

To delete an AWS CloudFormation stack for live streaming

1. Sign in to the AWS Management Console and open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation/.
2. Check the checkbox for the stack, and click **Delete Stack**.
3. Click **Yes, Delete** to confirm.
4. To track the progress of the stack deletion, check the checkbox for the stack, and click the **Events** tab in the bottom frame.

Frequently Asked Questions

- What is the price for Live Smooth Streaming using CloudFront? (p. 232)
- Can I deliver my live streaming video to both Smooth Streaming clients and Apple devices? (p. 232)
- How can I set-up a CNAME alias for my Amazon EC2 instance or my CloudFront distribution? (p. 232)
- How can I enable access to the Windows server? (p. 233)
- How can I securely connect to my Amazon EC2 instance running Windows IIS Media Services? (p. 233)
- How can I restrict access to my Live Smooth Streaming content from another domain? (p. 233)

What is the price for Live Smooth Streaming using CloudFront?

To Smooth Stream your live event, you pay only for the AWS resources you consume:

- For pricing information about Amazon EC2 instances running Windows Server, see **Pricing** on the Amazon EC2 Running Microsoft Windows Server & SQL Server page.
- For pricing information about CloudFront, see Amazon CloudFront Pricing.

There is no charge for using AWS CloudFormation.

Can I deliver my live streaming video to both Smooth Streaming clients and Apple devices?

Yes. You can use Microsoft Expression Encoder 4 Pro to encode your live video for both Smooth Streaming clients (for example, Microsoft Silverlight) and Apple devices (for example, iPad and iPhone). After your AWS CloudFormation stack is launched, you will find the manifest file URLs both for Live Smooth Streaming (.isml) and for Apple HLS (.m3u8) on the **Outputs** tab of your AWS CloudFormation template.

How can I set-up a CNAME alias for my Amazon EC2 instance or my CloudFront distribution?

Your Amazon EC2 Windows Server instance comes with an internal and an external DNS name. Amazon EC2 does not provide access to modify these DNS settings. If you want to map an existing domain name to your Amazon EC2 instance running Windows Server, use a DNS service provider such as Amazon Route 53. When using your own domain name, we recommend that you map to the instance's external DNS name using a CNAME, not by using an A record that points to the instance's IP address.

To map your own domain name to your CloudFront distribution, see Using Alternate Domain Names (CNAMEs) (p. 29).

How can I enable access to the Windows server?

To enable access to port 3389 on your Windows server via selected IP addresses

By default, the Amazon EC2 security group for your Windows server instance does not have port 3389 enabled; this is the port you use to administer the Windows server. If you want to log on to your Windows server instance, perform the following procedure to enable access via port 3389.

1. Sign in to the AWS Management Console and open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the **Region** list, click the Amazon EC2 region in which you used AWS CloudFormation to create your Amazon EC2 instance.
3. In the **Navigation** pane, click **Security Groups**.
4. In the **Security Groups** pane, click the row for which the value of the **Name** column begins with the name of the AWS CloudFormation stack that you created in Creating an AWS CloudFormation Stack for Live Smooth Streaming (p. 222).
5. In the bottom pane, click the **Inbound** tab.
6. To enable access to your Windows server and specify the client IP addresses that can access the server:

   a. In the **Create a new rule** list, do not change the default value, **Custom TCP rule**.
   b. In the **Port range** field, enter **3389**.
   c. In the **Source** field, enter an IP address or range, or enter the name of another security group. For more information, click **Help**.
   d. Click **Add Rule**.
   e. To create additional rules, repeat Steps a through d.
   f. Click **Apply Rule Changes**.

How can I securely connect to my Amazon EC2 instance running Windows IIS Media Services?

To connect to your Windows server instance, you must retrieve the initial password for the Administrator account and then use it with Windows Remote Desktop. You'll also need the contents of the private key file that you created, for example, *<keypairname.pem>*.pem. For more information, go to Getting Started Guide AWS Computing Basics for Windows.

How can I restrict access to my Live Smooth Streaming content from another domain?

Microsoft Silverlight includes support for cross-domain connectivity, which allows the Silverlight player to access content from locations other than the domain where the Smooth Streaming content originates. The security policy system in Silverlight requires that a Silverlight policy file named `ClientAccessPolicy.xml` be downloaded from a target domain before a network connection is allowed to access a network resource under that target domain. A default policy file is already included at the root of the default website on your Windows server running on Amazon EC2. To restrict cross-domain access, log on to your Windows server and update the `ClientAccessPolicy.xml` file.

Additional Documentation

Microsoft Documentation

- IIS Smooth Streaming Deployment Guide
- IIS Media Services 4.1 Readme
- IIS Smooth Streaming Management REST Services
- Configuring Authentication in IIS 7
- Microsoft Expression Encoder blog
- Managing live publishing points from Microsoft Expression Encoder 4 Pro SP2
- Live IIS Smooth Streaming in Expression Encoder 4 Pro
- Apple HTTP Live Streaming with IIS Media Services

Amazon Web Services Documentation

- Amazon EC2 Running Microsoft Windows Server & SQL Server
- Amazon Elastic Compute Cloud Microsoft Windows Guide
- Amazon CloudFront
- AWS CloudFormation

Live HTTP Streaming Using Wowza Streaming Engine 4.0

You can use Wowza Streaming Engine 4.0 to create live streaming sessions for global delivery using CloudFront. Wowza Streaming Engine 4.0 supports the following HTTP based streaming protocols:

- HLS (HTTP Live Streaming)
- HDS (HTTP Dynamic Streaming)
- Smooth Streaming

When a user streams a video using one of the above protocols, the video is broken into smaller chunks that are cached in the CloudFront network for improved performance and scalability.

This tutorial explains how to integrate CloudFront with Wowza Streaming Engine 4.0 running on an Amazon EC2 instance. For more information about how to manage and secure your Amazon EC2 instance, refer to the Amazon EC2 documentation. For more information about Wowza Streaming Engine options not covered in this tutorial, see the Wowza documentation.

Topics

- Creating an Amazon Web Services Account (p. 235)
- Creating an Amazon EC2 Key Pair (p. 236)
- Getting a License for Wowza Streaming Engine 4.0 (p. 236)
- Subscribing to Wowza Streaming Engine 4.0 through AWS Marketplace (p. 236)
- Creating an AWS CloudFormation Stack for Live Streaming (p. 237)
- Verifying that Wowza Streaming Engine 4.0 Is Running (p. 238)
- Setting Up an Encoder to Publish a Live Stream (p. 239)
- Playing the Live Stream in a Web Application (p. 240)
- Deleting an AWS CloudFormation Stack for Live Streaming (p. 241)
- Frequently Asked Questions (p. 241)
- Additional Documentation (p. 243)

Creating an Amazon Web Services Account

If you already have an AWS account, skip to Creating an Amazon EC2 Key Pair (p. 236). If you don't already have an AWS account, use the following procedure to create one.

To create an AWS account

1. Go to http://aws.amazon.com, and then click **Sign Up**.
2. Follow the on-screen instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.

Next: Creating an Amazon EC2 Key Pair (p. 236)

Creating an Amazon EC2 Key Pair

If you already have an Amazon EC2 key pair for the Amazon EC2 region in which you want to configure live streaming, skip to Getting a License for Wowza Streaming Engine 4.0 (p. 236). If you don't have a key pair in that region, follow the steps below.

A key pair is a security credential similar to a password and is specific to an AWS region. You need to specify a key pair when you create an AWS CloudFormation stack for live streaming, later in this process. After live streaming is configured, you use the key pair to securely connect to your Amazon EC2 instance.

To create an Amazon EC2 key pair

1. Sign in to the AWS Management Console and open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the region selector, click the region in which you want to create the key pair.

   You must create the key pair in the same region where you will create your AWS CloudFormation stack for live streaming later in this process. We recommend that you create the key pair and the stack for live streaming in the region that is closest to the location from which the live stream will be published.
3. In the left navigation pane, click **Key Pairs**.
4. In the key pairs pane, click **Create Key Pair**.
5. In the **Create Key Pair** dialog box, enter a name for the key pair, and make note of the name. You'll need it later when you create an AWS CloudFormation live-streaming stack.
6. Click **Create**, and, when prompted, save the `.pem` file to a safe place on your computer. Note that you will not be able re-download this file.
7. Click **Close** to close the **Create Key Pair** dialog box.

Next: Getting a License for Wowza Streaming Engine 4.0 (p. 236)

Getting a License for Wowza Streaming Engine 4.0

You need a license for Wowza Streaming Engine 4.0 to configure live streaming. To buy a license, go to the Licenses Built for You page on the Wowza website, select the licensing option that works best for you, and follow the on-screen instructions.

Next: Subscribing to Wowza Streaming Engine 4.0 through AWS Marketplace (p. 236)

Subscribing to Wowza Streaming Engine 4.0 through AWS Marketplace

The next step is to subscribe to Wowza Streaming Engine 4.0 in AWS Marketplace.

To order Wowza Streaming Engine for Amazon Web Services

1. Go to the Amazon Web Services page, and sign in using your Amazon.com account or create a new account.
2. Go to https://aws.amazon.com/marketplace/pp/B00IPTPABM. Review the details, and click **Continue**.
3. Click the **Launch with EC2 Console** tab.
4. Review the pricing information, and click **Accept Terms**.

Important
Don't use the buttons on this page to launch Wowza.

Next: Creating an AWS CloudFormation Stack for Live Streaming (p. 237)

Creating an AWS CloudFormation Stack for Live Streaming

The following procedure uses an AWS CloudFormation template to create a stack that launches the AWS resources required by live streaming, including an Amazon EC2 instance.

Important
You begin to incur hourly charges for an Amazon EC2 instance when you create the AWS CloudFormation stack that deploys that instance. Charges continue to accrue until you delete the AWS CloudFormation stack regardless of whether you use the Amazon EC2 instance to stream live video. When your live event is over, delete the stack that you created for live streaming. This deletes the AWS resources that were created for your live-streaming event, and stops the AWS charges for the resources. For more information, see Deleting an AWS CloudFormation Stack for Live Streaming (p. 241).

To create an AWS CloudFormation stack for live streaming

1. To start the wizard, click one of the following Amazon EC2 regions:

   - Create a stack in US East (Northern Virginia)
   - Create a stack in US West (Oregon)
   - Create a stack in US West (Northern California)
   - Create a stack in EU (Ireland)
   - Create a stack in Asia Pacific (Singapore)
   - Create a stack in Asia Pacific (Sydney)
   - Create a stack in Asia Pacific (Tokyo)
   - Create a stack in South America (Sao Paulo)

2. If you are not already signed in to the AWS Management Console, sign in when prompted. The wizard starts, and the selected URL automatically appears under **Provide an S3 URL to template**.
3. (Optional) In the **Create a New Stack** wizard, you can change the stack name to something appropriate for your live-streaming event. The stack name must not contain spaces and must be unique within your AWS account.

   Do not change the **Template** option or the address in **Provide an S3 URL to template**.
4. Click **Next Step**.
5. On the Specify Parameters page, for **ApplicationName**, enter a short name (without spaces) for your Wowza application, or keep the default.
6. For **InstanceType**, enter an instance type, which determines pricing for your Wowza instance. For more information about Amazon EC2 instance types, see Available Instance Types.

   For information about pricing, see Amazon EC2 Pricing.
7. For **KeyPair**, enter the name of an Amazon EC2 key pair for the region in which you want to create the live streaming stack. The key pair must be associated with the account that you're currently logged on with. If you created a key pair when you performed the procedure in Creating an Amazon EC2 Key Pair (p. 236), enter that name here.

8. For **StartupPackageURL**, enter a URL that points to a startup package to configure the Wowza Streaming Engine to your needs, or keep the default.
9. For **StreamName**, enter a short name (without spaces) for your live stream, or keep the default.
10. For **WowzaLicenseKey**, enter the license key you got when you performed the procedure in the Getting a License for Wowza Streaming Engine 4.0 (p. 236) topic. If you purchased AddOns, you can include additional license keys by separating the key values with a pipe (|) character.
11. Click **Next Step**.
12. (Optional) On the Options page, add the key-value pairs for any tags you plan to use. See Adding Tags to Your AWS CloudFormation Stack for more information about using tags.
13. (Optional) To configure SNS notification, to specify how long you're willing to wait for the stack to be created, to choose whether to roll back changes if stack creation fails, and to enter a stack policy, click **Advanced**, and adjust settings as desired.
14. Click **Next Step**
15. Review the settings for the stack. When you're satisfied, click **Create**, and AWS CloudFormation creates the stack.

    Creating your stack creation may take several minutes. To track the progress of stack creation, select the stack, and click the **Events** tab. If AWS CloudFormation cannot create the stack, the **Events** tab lists error messages.

    When your stack is ready, in the list of stacks, the status for the stack changes to **CREATE_COMPLETE**.

    When your stack is created, click the **Outputs** tab, which displays the stack creation outputs. You will use these values when you set up the encoder later in the process.

Next: Verifying that Wowza Streaming Engine 4.0 Is Running (p. 238)

Verifying that Wowza Streaming Engine 4.0 Is Running

After AWS CloudFormation creates the stack, follow the steps below to verify that Wowza Streaming Engine 4.0 is running on the Amazon EC2 instance that you provisioned via AWS CloudFormation.

To verify that Wowza Streaming Engine 4.0 is running

1. Sign in to the AWS Management Console and open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation/.
2. In the region selector, click the region in which you created the AWS CloudFormation stack.
3. In the list of stacks, select the stack that you created in Creating an AWS CloudFormation Stack for Live Streaming (p. 237).
4. Click the **Outputs** tab.
5. On the **Outputs** tab, get the value of the **WowzaServerLoginInfo** key, which you'll use for login credentials in the next step.
6. Click the URL in the **WowzaEngineManagerURL** key, for example, **http://ec2-xx-xx-xxx-xxx.compute-1.amazonaws.com:8088/enginemanager**. When you're prompted for login credentials, use the value of the **WowzaServerLoginInfo** key that you got in step 5.

    Important
    This URL uses port 8088 to connect to the Amazon EC2 instance that is running Wowza Streaming Engine. Depending on your firewall settings, you might not be able to connect to the Amazon EC2 instance. If you have trouble, contact your network administrator.

Next: Setting Up an Encoder to Publish a Live Stream (p. 239)

Setting Up an Encoder to Publish a Live Stream

You need to encode the live stream captured by your device before you send it to Wowza Streaming Engine 4.0. You can encode the live stream by using either the Wowza GoCoder app for iOS-based devices or encoders that support RTMP encoding, such as Telestream Wirecast.

The steps for publishing a stream from your encoder to Wowza Streaming Engine vary with your choice of encoder. For more information about how to configure your encoder, go to Specific Encoding Technologies on the Wowza website or review the documentation for your encoder.

Encode Apple HLS streams using the following formats:

- **Video:**
  - **Apple iPhone, iPod, and iPod touch** – H.264 Baseline Profile Level 3.0. Don't use B-frames when targeting iPhone and iPod devices.
  - **Apple iPad** – H.264 Main Profile Level 3.1
- **Audio:** AAC-LC up to 48 kHz, stereo audio.

Smooth Streams must have both audio and video. The key frame frequency must be between 1 and 4 seconds. We recommend a key frame frequency of two seconds.

Wowza Gocoder

To configure Wowza GoCoder to publish a live stream, take the following steps.

> **Note**
> Wowza GoCoder is available for purchase from the Apple AppStore.

1. Go to the Wowza tutorial How to use Wowza GoCoder video broadcasting iOS app with Wowza Streaming Engine.
2. Follow the procedure in the Configure Wowza GoCoder App section of the tutorial. Specify the following values for **Host** settings:

   a. Open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation/.
   b. In the AWS CloudFormation console, click the **Outputs** tab.
   c. Copy the value of the **WowzaEngineDomainName** key, for example, **ec2-*xx-xx-xxx-xxx*.compute-1.amazonaws.com**.
   d. For **Server**, paste the value that you just copied.
   e. For **Port**, enter **1935**.

3. Specify the following values for **Application** settings:

   a. For **Application**, enter the application name that you specified when you created the stack, for example, `livecf`.
   b. On the **Outputs** tab of the AWS CloudFormation console, copy the value of the **PublishStreamName** key, for example, **myStream**.
   c. For **StreamName**, paste the value that you just copied in the previous step.

4. Change other values as applicable.

RTMP encoder

RTMP encoders typically use the following settings:

Publish URL
This is the value of the AWS CloudFormation **PublishRTMPURL** key, for example, **rtmp://ec2-*xx-xx-xxx-xxx*.compute-1.amazonaws.com/livecf**.

Stream Name
This is the value of the AWS CloudFormation **PublishStreamName** key, for example, **myStream**.

Login Credentials
If you are prompted for login credentials, use the values from the AWS CloudFormation **Wowza-ServerLoginInfo** key, for example, **username=wowza, password=i-1234a567**.

Next: Playing the Live Stream in a Web Application (p. 240).

Playing the Live Stream in a Web Application

Wowza Media Services provides online example player web pages that you can use to play the live stream from your Wowza Streaming Engine distribution. These players can help you verify that your streaming stack has been set up correctly. You can use the same streaming manifest URLs for other players that support the streaming protocol you want to use.

Perform the applicable procedure to get the embed code that you will include in your web page for the live stream:

> **Note**
> Wait at least 30 seconds after you perform the applicable procedure in Setting Up an Encoder to Publish a Live Stream (p. 239) before following the steps below to play the stream.

- To play your live HDS stream on Adobe Flash Player via CloudFront (p. 240)
- To play your live HLS stream on an Apple or other device via CloudFront (p. 240)
- To play your Live Smooth Stream via CloudFront (p. 241)

To play your live HDS stream on Adobe Flash Player via CloudFront

1. Open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation/.
2. Select the stack for live HTTP streaming.
3. In the AWS CloudFormation console, click the **Outputs** tab.
4. Copy the value of the **PlaybackHDSURL** key, for example, **http://d111111abcdef8.cloudfront.net/livecf/myStream/manifest.f4m**.
5. Go to the Flash HTTP Player example web page on the Wowza website, paste the URL that you copied in the previous step into the **Stream** field, and click **Connect**.

To play your live HLS stream on an Apple or other device via CloudFront

1. Open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation/.
2. Select the stack for live streaming.
3. In the AWS CloudFormation console, click the **Outputs** tab.
4. Copy the value of the **PlaybackHLSURL** key, for example, **http://d111111abcdef8.cloudfront.net/livecf/myStream/playlist.m3u8**.
5. Using one of the following applications, go to the iOS/Mac OS X example web page on the Wowza website, paste the URL that you copied in the previous step into the **Stream** field, and click **Connect**:

- Safari web browser on a computer running Mac OS X Snow Leopard (version 10.6) or later
- QuickTime Player 10.x or later on a computer running Mac OS X Snow Leopard (version 10.6) or later
- Safari web browser on an Apple iOS device

To play your Live Smooth Stream via CloudFront

1. Open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation/.
2. Select the stack for live streaming.
3. In the AWS CloudFormation console, click the **Outputs** tab.
4. Copy the value of the **PlaybackSmoothURL** key, for example, **http://d111111abcdef8.cloudfront.net/livecf/myStream/Manifest**.
5. Go to the Silverlight Player example web page on the Wowza website, paste the URL that you copied in the previous step into the **Stream** field, and click **Connect**.

Next: Deleting an AWS CloudFormation Stack for Live Streaming (p. 241).

Deleting an AWS CloudFormation Stack for Live Streaming

When your live event is over, delete the stack that you created for live streaming. This deletes the AWS resources that were created for your live-streaming event and stops the on-demand charges for the resources.

To delete an AWS CloudFormation stack for live streaming

1. Open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation/.
2. In the upper right corner, click the region in which you created your stack.
3. Select the stack, and click **Delete Stack**.
4. Click **Yes, Delete** to confirm.
5. To track the progress of the stack deletion, select the stack, and click the **Events** tab.

Frequently Asked Questions

- What is the price for live HTTP streaming using CloudFront and Wowza Streaming Engine 4.0? (p. 242)
- How can I use Secure Shell (SSH) to connect to my Amazon EC2 instance that is running Wowza Streaming Engine 4.0? (p. 242)
- How can I create a CNAME alias for my Amazon EC2 instance or for my CloudFront distribution? (p. 242)
- Can I stream my live event to Flash Player–compatible devices, Apple devices, and Smooth Streaming players at the same time? (p. 242)
- Does Wowza Streaming Engine 4.0 support HTML5? (p. 242)
- Can I serve private live streams using Wowza and CloudFront? (p. 243)

What is the price for live HTTP streaming using CloudFront and Wowza Streaming Engine 4.0?

Charges for live HTTP streaming using CloudFront and Wowza Streaming Engine 4.0 include:

- **Wowza Streaming Engine software and AddOns:** For more information, see Licenses Built For You on the Wowza website.
- **Amazon EC2:** For more information, see the **Linux** tab in the On-Demand Instances Pricing table.
- **CloudFront:** For more information, see Amazon CloudFront Pricing.

There is no charge for using AWS CloudFormation.

How can I use Secure Shell (SSH) to connect to my Amazon EC2 instance that is running Wowza Streaming Engine 4.0?

You can use SSH to connect to your Amazon EC2 instance in just a few steps.

To use SSH to connect to your Amazon EC2 instance that is running Wowza Streaming Engine 4.0

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigation, click **Instances**.
3. Right-click the correct instance, and click **Connect** to view instructions on how to use SSH to connect to your Amazon EC2 instance. The username is **ec2-user**.

How can I create a CNAME alias for my Amazon EC2 instance or for my CloudFront distribution?

Your Amazon EC2 instance running Wowza Streaming Engine 4.0 comes with an internal and an external DNS name. Amazon EC2 does not provide access to modify these DNS settings. If you want to map an existing domain name to your Amazon EC2 instance running Wowza Streaming Engine, use a DNS service provider such as Amazon Route 53. When using your own domain name, we recommend that you map to the instance's external DNS name using a CNAME, not by using an A record that points to the instance's IP address.

To map your own domain name to your CloudFront distribution, see Using Alternate Domain Names (CNAMEs) (p. 29).

Can I stream my live event to Flash Player–compatible devices, Apple devices, and Smooth Streaming players at the same time?

Yes, Wowza Streaming Engine 4.0 enables the delivery of live streams in Adobe HTTP Dynamic Streaming (Adobe HDS), Apple HTTP Live Streaming (Apple HLS), and Microsoft Smooth Streaming formats simultaneously, which enables playing on Adobe Flash Player applications, Apple iOS devices, and Smooth Streaming players, respectively.

Does Wowza Streaming Engine 4.0 support HTML5?

Yes, Wowza Streaming Engine can deliver content to HTML5 in the following configurations:

- You can use the Apple HLS streaming format on Apple iOS devices.
- You can use the Smooth Streaming format on Windows 8 devices. For more information, go to Walkthrough: Building Your First HTML5 Smooth Streaming Player on the MSDN website.
- For other browsers supporting HTML5, you can use Wowza Streaming Engine to deliver video via progressive download.

Can I serve private live streams using Wowza and CloudFront?

At this time, live streams cannot be delivered securely using CloudFront signed URLs. However, progressively downloaded media can be delivered privately by using signed URLs. For more information, see Serving Private Content through CloudFront (p. 118).

Additional Documentation

The following resources may assist you as you work with Wowza.

Wowza Documentation

The Wowza website includes articles, documentation and pricing information for using Wowza live streams with Amazon Web Services:

- Wowza for Amazon EC2 Articles
- Wowza Streaming Engine User's Guide
- Wowza on Amazon Web Services

Amazon Web Services Documentation

The following resources include user guides and reference works for Amazon Web Services.

- Amazon Elastic Compute Cloud documentation
- AWS CloudFormation documentation

Restricting Access to Files in a CloudFront Distribution Based on Geographic Location (Geoblocking)

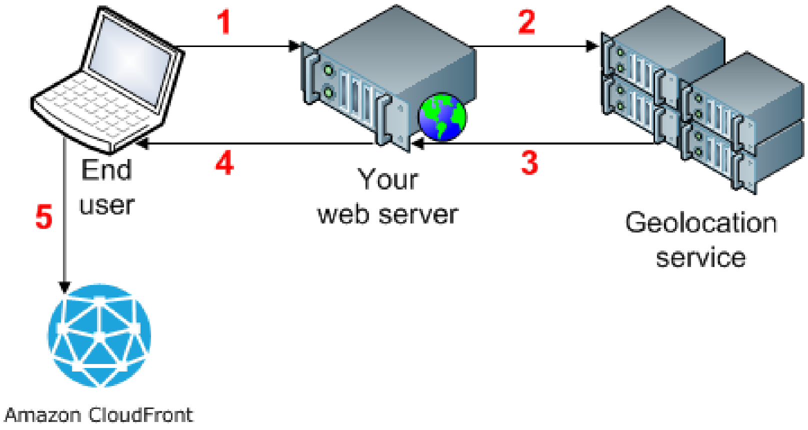
Topics

- Creating an Amazon Web Services Account (p. 246)
- Sample Code for Digital Element (p. 246)
- Sample Code for MaxMind (p. 256)
- Frequently Asked Questions (p. 266)
- Additional Services and Documentation (p. 267)

Amazon CloudFront improves the performance, reliability, and availability of your websites and applications by distributing your web content, such as images, video, and audio to a worldwide network of edge locations. When an end user requests your content, CloudFront serves your content to the user from the edge location that has the lowest latency for that user at that moment. If you have geographic restrictions on where your content can be distributed, you can use CloudFront with a third-party geolocation service to control distribution of your content according to the location of a request. This is known as geo restriction or geoblocking. For example, if a request comes from a country where, for copyright reasons, you are not authorized to distribute your content, you can block the request and direct the requester to a message that explains the situation.

Here's how it works:

1. An end user who is viewing your website requests a web page or a file that is geo restricted.
2. Your web application gets the end user's IP address from the request and sends the IP address to a geolocation service. You will need an account with one of these services.
3. The geolocation service determines the geographic location of the end user's IP address and returns the result to your web application.
4. Your web application compares the end user's location with a list of locations where the file can (or can't) be distributed:
   - If the end user is allowed to access the web page or file, your application creates a CloudFront signed URL and returns it to the end user.
   - If the end user is not allowed to access the web page or file, your web application returns the URL of a "you are not authorized" message to the end user.
5. If the end user is allowed to access the web page or file, the end user's browser automatically uses the signed URL to request the file from CloudFront.



Using CloudFront and a third-party geolocation service to restrict access to your content from your application layer gives you full control over your end user's experience. For end users whose access is blocked, your application can display a meaningful message instead of returning an error code. You can also customize the error message you display for your end users according to their location.

> **Note**
> If you're restricting distribution of your content in geographic regions that follow country boundaries and if you want to restrict distribution of all of the files associated with your distribution (instead of individual files), you might prefer to use CloudFront geo restriction. For more information, see Restricting the Geographic Distribution of Your Content (p. 56).

The following task list guides you through the process of implementing geoblocking functionality in your applications to restrict access to the content in your CloudFront distribution according to the end user's location.

Task list for restricting access to files in a CloudFront distribution based on geographic location

1. Get an account with a geolocation service.

   This section provides sample code for Digital Element and for MaxMind, but any geolocation service is supported.
2. If you don't already have an AWS account, create one. For more information, see Creating an Amazon Web Services Account (p. 246).
3. Upload your content to an Amazon Simple Storage Service (S3) bucket. For more information, see the Amazon S3 documentation.
4. Configure Amazon CloudFront and Amazon S3 to serve private content. For more information, see Serving Private Content through CloudFront (p. 118).
5. Write your web application to do the following:

   a. Send the IP address for each end-user request to the geolocation service.

   b. Evaluate the return value from the geolocation service (commonly a country code) to determine whether the end user is in a location to which you want CloudFront to distribute your content.

   c. Either generate a signed URL for your CloudFront content, or block access to the content.

   Java, .NET, and PHP sample code is provided below for Digital Element and for MaxMind. See the applicable topic:

   - Sample Code for Digital Element (p. 246)

- Sample Code for MaxMind (p. 256)

If you're using another geolocation service, refer to their documentation.

Amazon Web Services provides SDKs for Java, .NET, and PHP. For more information, see the applicable page on the Amazon Web Services website:

- Java Developer Center
- Windows & .NET Developer Center
- PHP Developer Center

Creating an Amazon Web Services Account

Note
When you create an account, AWS automatically signs up the account for all services. You are charged only for the services you use.

To create an AWS account

1. Go to http://aws.amazon.com, and click **Create an AWS Account**.
2. Follow the on-screen instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.

Sample Code for Digital Element

The samples in this section show how to get a location from Digital Element from an end user's IP address. The samples also show how to create a signed URL for a requested object if you are allowed to distribute the content to the end user's location.

All sample code was tested before the document was published, but subsequent changes to the Digital Element API could affect whether the samples are still accurate. For the latest information, go to the Digital Element documentation.

See the applicable sample code:

- Java Sample Code for Digital Element (p. 246)
- .NET Sample Code for Digital Element (p. 251)
- PHP Sample Code for Digital Element (p. 254)

Note
In the code examples, red italicized text is a placeholder. Replace this text with whatever values are appropriate for your situation.

Java Sample Code for Digital Element

The code example provided here obtains the country code that is associated with an end user's IP address and allows the user to access CloudFront content if the user is in a location where distribution is allowed. For the purposes of the example, the program is authorized to distribute the requested content to any country except Australia (country code AU).

GetCountryCodeServlet.java

GetCountryCodeServlet.java calls GetDigitalElementCountryCode.java, which is shown later in this article, to ask Digital Element for the country code that is associated with an end user's IP address. If the country code is not AU (Australia), GetCountryCodeServlet.java calls SignedUrl.java to create a signed URL that the end user can use to access a file in the CloudFront distribution.

```
/*
 * Copyright 2011 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *  http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

// Signed URLs for a private distribution
// Note that Java supports SSL certificates only in DER format,
// so you will need to convert your PEM-formatted file to DER format.
// To do this, you can use openssl:
// openssl pkcs8 -topk8 -nocrypt -in origin.pem -inform PEM -out new.der -outform
 DER
// For the encoder to work correctly, you should also add the
// bouncy castle jar to your project and then add the provider.ds.

import java.io.IOException;
import java.io.PrintWriter;
import java.util.StringTokenizer;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class GetCountryCodeServlet extends HttpServlet {
   private static final long serialVersionUID = 1L;

   final String GEOAPIURL = "Digital Element URL";
   final String GEOAPITOKEN = "Digital Element user token";
   final String PATHTODER = "path to .der file";
   final String KEYPAIRID ="CloudFront key pair ID";
   final String HTTPORHTTPS = "https";
   final String CFDISTRIBUTION = "dxxxx.cloudfront.net";
   final String CFPATH = "CloudFront URL for file";
   // date and time that CloudFront's signed URL expires,
   // in Coordinated Universal Time
   final String EXPIRETS = "2012-11-14T22:20:00.000Z";
   final String BLOCKEDCOUNTRY="AU";

   protected void doGet(HttpServletRequest request, HttpServletResponse response)
 throws ServletException, IOException {
      String ip = null;
      StringTokenizer st = null;
      PrintWriter out = response.getWriter();

      String headers = request.getHeader("X-FORWARDED-FOR");

      if (headers!= null){
         st = new StringTokenizer(headers,",");

         while (st.hasMoreTokens()) {
            ip = st.nextToken();
         }
      }

      //Get the client's IP addr in case X-Forwarded-IP header doesn't exist

      if (ip == null) ip = request.getRemoteAddr();

      try {
        GetDigitalElementCountryCode country = new GetDigitalElementCountryCode(
 GEOAPIURL,GEOAPITOKEN );

         if ( !country.getCountry(ip).equalsIgnoreCase(BLOCKEDCOUNTRY)){

            SignedUrl myApp = new SignedUrl(KEYPAIRID,PATHTODER);
          out.println(myApp.getSignedHash(HTTPORHTTPS,CFDISTRIBUTION,CFPATH,EX
PIRETS));

         }else {
            out.println("You cannot access this link.");
         }
      } catch (Exception e1) {
         e1.printStackTrace();
      }
   }
}
```

GetDigitalElementCountryCode.java

GetDigitalElementCountryCode.java sends Digital Element a request that includes an end user's IP address. The return value is a country code.

```
/*
 * Copyright 2011 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *  http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */
```

```
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.NodeList;

public class GetDigitalElementCountryCode {

   private static String geoApiEndPoint;
   private static String apiToken;

   GetDigitalElementCountryCode(String mygeoApiEndPoint, String myapiToken){
      geoApiEndPoint = mygeoApiEndPoint;
      apiToken = myapiToken;
   }

    public String getCountry(String enduserIP) throws Exception {

      String geoApiURL = "http://"+geoApiEndPoint+"?u="+apiToken+"&ip="+end
userIP;

      DocumentBuilderFactory docBuilderFactory = DocumentBuilderFactory.newIn
stance();
      DocumentBuilder docBuilder = docBuilderFactory.newDocumentBuilder();
      Document doc = docBuilder.parse(geoApiURL);
      // normalize text representation
      doc.getDocumentElement ().normalize ();

      NodeList listOfPersons = doc.getElementsByTagName("response");
      Element el = (Element)listOfPersons.item(0);
      String country = el.getAttribute("edge-two-letter-country");

      return country;
   }
}
```

SignedUrl.java

SignedUrl.java creates a signed URL that the end user can use to access a file in the CloudFront distribution.

```
/*
 * Copyright 2011 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *  http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */
```

```
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.security.Security;
import java.text.ParseException;

import org.jets3t.service.CloudFrontService;
import org.jets3t.service.CloudFrontServiceException;
import org.jets3t.service.utils.ServiceUtils;

public class SignedUrl {
   // Signed URLs for a private distribution
   // Note that Java supports SSL certificates only in DER format,
   // so you need to convert your PEM-formatted file to DER format.
   // To do this, you can use openssl:
   // openssl pkcs8 -topk8 -nocrypt -in origin.pem -inform PEM -out new.der -
outform DER
   // For the encoder to work correctly, you should also add the
   // bouncy castle jar to your project and then add the provider.ds.

   private static String keyPairId;
   private static String privateKeyFilePath;

   SignedUrl(String mykeyPairId, String myprivateKeyFilePath){
      keyPairId = mykeyPairId;
      privateKeyFilePath = myprivateKeyFilePath;
   }

   public String getSignedHash(String protocol, String cfDistribution, String
objectUri, String expTime) throws FileNotFoundException, IOException,
   CloudFrontServiceException, ParseException{

   Security.addProvider(new org.bouncycastle.jce.provider.BouncyCastlePro
vider());

   // Convert your DER file into a byte array.

     byte[] derPrivateKey = ServiceUtils.readInputStreamToBytes(new FileInput
Stream(privateKeyFilePath));

   // Generate a "canned" signed URL to allow access to a
   // specific distribution and object

   String signedUrlCanned = CloudFrontService.signUrlCanned(
       protocol+ "://" + cfDistribution + "/" + objectUri, // Resource URL or
Path
       keyPairId,     // Certificate identifier,
                      // an active trusted signer for the distribution
       derPrivateKey, // DER Private key data
       ServiceUtils.parseIso8601Date(expTime) // DateLessThan
       );

   return signedUrlCanned;
   }
}
```

.NET Sample Code for Digital Element

The following sample application gets the IP address of the end user and sends the IP address to Digital Element. Digital Element returns the country code (in XML format) that corresponds with the end user's IP address. The application parses the XML and evaluates whether the value returned by Digital Element matches the blocked country code. If the end user's country is blocked, the application displays a message to that effect. If the end user's country is not blocked, the application creates a signed URL that expires in one minute, performs the substitutions necessary to ensure that the URL doesn't include any invalid characters, and redirects the user's browser to the signed URL.

```
<%@ Page Language="C#" AutoEventWireup="true"  %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "ht
tp://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
   <div>
   <%=GetContent()%>
   </div>
   </form>
</body>
</html>

<%@ Import Namespace="System.Linq" %>
<%@ Import Namespace="System.Xml.Linq" %>
<%@ Import Namespace="System.Security.Cryptography" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>

<script runat="server">

   // Key pair ID for the CloudFront key pair
   private const string KEYPAIR_ID = "CloudFront key pair ID";

   // Private key for the CloudFront key pair.
   // The value is derived from opensslkey.
   private const string PRIVATE_KEY = "private key";

   // JSON policy statement used in the expiring URL
   private const string POLICY = "{{\"Statement\":[{{\"Resource\":\"{0}\",\"Con
dition\":{{\"DateLessThan\":{{\"AWS:EpochTime\":{1}}}}}}]}}";

   // Digital Element user token to be passed to geolocation service call

   private const string USERTOKEN = "Digital Element user token";
   private const string GEOAPIURL = "Digital Element URL";

   // GEO IP service URL with parameters:
   // {0} = User Token and {1} = IP Address
   private const string SERVICEURL = GEOAPIURL + "?u={0}&ip={1}";

   // Array of countries to block
   private static readonly string[] COUNTRIES_TO_BLOCK = new String[] {"US"};

   private const string BLOCKED_MSG = "Your access to this content is blocked
because you're visiting from '{0}'.";

   /// <summary>
   /// Returns the IP address coming from the request object.
   /// </summary>
   /// <returns>The IP address for the request.</returns>
   private string GetOriginIpAddress()
   {
      // .NET provides Request.UserHostAddress to get the
      // remote IP address, but this could be the IP address of the
      // last proxy in a chain, for example, an Elastic Load Balancer.
      // Instead, use the HTTP_X_FORWARDED_FOR header if one exists.
     string forwardedIpAddresses = this.Request.ServerVariables["HTTP_X_FORWAR
DED_FOR"];

      if (string.IsNullOrEmpty(forwardedIpAddresses))
      {
         // Simply return the UserHostAddress.
         return Request.UserHostAddress;
      }
      else
      {
         // Get the last item in the list.
         return forwardedIpAddresses.Split(',').Last().Trim();
      }
   }

   /// <summary>
   /// This function returns the country code
   /// associated with the IP address in the request object.
   /// </summary>
   /// <returns>The country code for the request.</returns>
   private string GetCountryCodeFromIP()
   {
      var ipAddress = GetOriginIpAddress();
      var serviceURL = String.Format(SERVICEURL, Server.UrlEncode(USERTOKEN),
Server.UrlEncode(ipAddress));

      try
      {
         var xDoc = XDocument.Load(serviceURL);
         var res = (from w in xDoc.Descendants("response") select w).First();

         return res.Attribute("edge-two-letter-country").Value.ToUpper();
      }
      catch(Exception ex)
      {
         // There was an error in making the web request.
         this.Response.Write(serviceURL +  "<br><br>");
         this.Response.Write(ex.Message);
         this.Response.End();
      }
      return null;
   }
   /// <summary>
   /// This function returns a signed URL that will expire in 1 minute.
   /// For more information, see "Create a URL Signature Using C# and the
   /// .NET Framework" in the Amazon CloudFront Developer Guide:
   /// http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Cre
ateSignatureInCSharp.html
   /// </summary>
   /// <param name="resourceUrl"></param>
   /// <returns></returns>
   private string GetSignedURL(string resourceUrl)
   {
      // Compute expiration date.
      var endTimeSpanFromNow = new TimeSpan(0, 1, 0);
      var intervalEnd = (DateTime.UtcNow.Add(endTimeSpanFromNow)) - new Date
Time(1970, 1, 1, 0, 0, 0, DateTimeKind.Utc);
      var endTimestamp = (int)intervalEnd.TotalSeconds; // Timestamp must be a
 whole number
      var expires = endTimestamp.ToString();
      var strPolicy = string.Format(POLICY, resourceUrl, expires);

      // Encrypt the policy.
      var bufferPolicy = Encoding.ASCII.GetBytes(strPolicy);
      var cryptoSHA1 = new SHA1CryptoServiceProvider();
      bufferPolicy = cryptoSHA1.ComputeHash(bufferPolicy);
      var providerRSA = new RSACryptoServiceProvider();
      providerRSA.FromXmlString(PRIVATE_KEY);
      var rsaFormatter = new RSAPKCS1SignatureFormatter(providerRSA);
      rsaFormatter.SetHashAlgorithm("SHA1");
      var signedPolicyHash = rsaFormatter.CreateSignature(bufferPolicy);
      var strSignedPolicy = System.Convert.ToBase64String(signedPolicyHash);

      // Build the query string with the expiration, policy signature,
      // and CloudFront key pair ID.
      var queryString = "Expires={0}&Signature={1}&Key-Pair-Id={2}";
      queryString = String.Format(queryString, Server.UrlEncode(expires),
Server.UrlEncode(strSignedPolicy), Server.UrlEncode(KEYPAIR_ID));
      var urlString = resourceUrl + "?" + queryString;
      return urlString;
   }

   /// <summary>
   /// Return a message saying this is blocked because of your country, or
   /// return an image tag.
   /// </summary>
   /// <returns></returns>
   public string GetContent()
   {
      var country = GetCountryCodeFromIP();
      if (COUNTRIES_TO_BLOCK.Contains(country))
      {
         // The country returned from the call to the geolocation service
         // is listed in the array of blocked countries.
         return string.Format(BLOCKED_MSG, country);
      }
      else
      {
         // The country returned from the call to the geolocation service
         // is NOT listed in the array of blocked countries
         // Get a CloudFront signed URL for the content and display it.
         var url = GetSignedURL("CloudFront URL");
         var img = "<img src='{0}' />";
         return String.Format(img, url);
      }
   }
</script>
```

PHP Sample Code for Digital Element

The following sample application gets the IP address of the end user and sends the IP address to Digital Element. Digital Element returns the country code (in XML format) that corresponds to the end user's IP address. The application then parses the XML, displays the country code that is blocked, and evaluates whether the value returned by Digital Element matches the blocked country code. If the end user's country is not blocked, the application displays a "You are not blocked" message, uses a canned policy to create a signed URL that expires in five minutes, performs the substitutions necessary to ensure that the URL doesn't include any invalid characters, and redirects the user's browser to the signed URL. If the end user's country is blocked, the application displays a "You are blocked" message and a graphic.

```
<!DOCTYPE html>
<html>
<head>
   <title>Geoblocking Test</title>
</head>
<body>
   <h1>Geoblocking Test</h1>

<?php
// Configure the private key (make sure this information is secure).
$private_key_filename = 'path to private key';
$key_pair_id          = 'CloudFront key pair ID';

/*
 * Configure the geoblocking parameters. The following variables
 * describe the two-letter country to be blocked, the
 * CloudFront URL for the file that you want to secure,
 * and the expiry time of the URL. Change these values as needed.
 */
$blocked_geo = 'uk';
$asset_path  = 'CloudFront URL for the object';
$expires     = time() + 300; // (5 minutes from now)

// Configure the URL to the geoblocking service.
$token       = 'Digital Element user token';
$address     = 'Digital Element URL';
$remote_ip   = get_remote_ip_address();
$service_url = $address . '?u=' . $token . '&ip=' . $remote_ip;

// Call the web service using the configured URL.
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $service_url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
$ws_response = curl_exec($ch);

// Parse the response with SimpleXML and get the geoblocking value.
$xml      = new SimpleXMLElement($ws_response);
```

```
$edge_geo = $xml->response->attributes()->{'edge-two-letter-country'};

echo '<p>The country being blocked is: ' . strtoupper($blocked_geo) . '</p>';

if ($edge_geo != $blocked_geo)
{
   echo '<p>Your country is: ' . strtoupper($edge_geo) . '</p>';
   echo '<p>You are not blocked.</p>';
   $signed_url = create_signed_url($asset_path, $private_key_filename,
$key_pair_id, $expires);
   echo '<img src="' . $signed_url . '" width="600" height="401" ' ;
}
else
{
   echo '<p>Your country is: ' . strtoupper($edge_geo) . '</p>';
   echo '<p>You are blocked.</p>';
   $blocked_url = 'http://s3.amazonaws.com/<Amazon S3 bucket>/blocked-image.jpg';

   echo '<img src="' . $blocked_url . '" alt="Access blocked" width="600"
height="401" ';
}

// Function definitions


function get_remote_ip_address()
{
   // Check to see if an HTTP_X_FORWARDED_FOR header is present.

   if($_SERVER["HTTP_X_FORWARDED_FOR"])

   {

      // If the header is present, use the last IP address.
      $temp_array      = explode(',', $_SERVER['HTTP_X_FORWARDED_FOR']);
      $temp_ip_address = $temp_array[count($temp_array) - 1];
   }
   else
   {
      // If the header is not present, use the
      // default server variable for remote address.
      $temp_ip_address = $_SERVER['REMOTE_ADDR'];
   }

   return $temp_ip_address;
}

function create_signed_url($asset_path, $private_key_filename, $key_pair_id,
$expires)
{
   // Build the policy.
   $canned_policy = '{"Statement":[{"Resource":"' . $asset_path
      . '","Condition":{"DateLessThan":{"AWS:EpochTime":'. $expires . '}}}]}';


   // Sign the policy.
   $signature = rsa_sha1_sign($canned_policy, $private_key_filename);
   // Make the signature is safe to be included in a URL.
   $encoded_signature = url_safe_base64_encode($signature);

   // Combine the above into a properly formed URL name.
   $temp_signed_url = $asset_path . '?Expires=' . $expires . '&Signature='
      . $encoded_signature . '&Key-Pair-Id=' . $key_pair_id;

   return $temp_signed_url;
}

function rsa_sha1_sign($policy, $private_key_filename)
{
   $signature = '';

   // Load the private key.
   $fp = fopen($private_key_filename, 'r');
   $private_key = fread($fp, 8192);
   fclose($fp);

   $private_key_id = openssl_get_privatekey($private_key);

   // Compute the signature.
   openssl_sign($policy, $signature, $private_key_id);

   // Free the key from memory.
   openssl_free_key($private_key_id);

   return $signature;
}

function url_safe_base64_encode($value)
{
   $encoded = base64_encode($value);

   // Replace the characters that cannot be used in a URL.
   return str_replace(array('+', '=', '/'), array('-', '_', '~'), $encoded);
}
?>

</body>
</html>
```

Sample Code for MaxMind

The samples in this section show how to get a location from MaxMind from an end user's IP address and, if you are authorized to distribute the requested object to the user's location, how to create a signed URL for the object.

All sample code was tested before the document was published, but subsequent changes to the MaxMind API could affect whether the samples are still accurate. For the latest information, go to the MaxMind documentation.

See the applicable sample code:

- Java Sample Code for MaxMind (p. 257)
- .NET Sample Code for MaxMind (p. 263)

- PHP Sample Code for MaxMind (p. 261)

Java Sample Code for MaxMind

GetCountryCodeServlet.java

GetCountryCodeServlet.java calls GetMaxMindCountryCode.java, which is shown later in this article, to ask MaxMind for the country code that is associated with an end user's IP address. If the country code is not AU (Australia), GetCountryCodeServlet.java calls SignedUrl.java to create a signed URL that the end user can use to access a file in the CloudFront distribution.

```
/*
 * Copyright 2011 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *  http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

// Signed URLs for a private distribution
// Note that Java supports SSL certificates only in DER format,
// so you will need to convert your PEM-formatted file to DER format.
// To do this, you can use openssl:
// openssl pkcs8 -topk8 -nocrypt -in origin.pem -inform PEM -out new.der -outform
 DER
// For the encoder to work correctly, you should also add the
// bouncy castle jar to your project and then add the provider.ds.

import java.io.IOException;
import java.io.PrintWriter;
import java.util.StringTokenizer;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class GetCountryCodeServlet extends HttpServlet {
   private static final long serialVersionUID = 1L;

   final String GEOAPIURL = "MaxMind URL";
   final String GEOAPITOKEN = "MaxMind user token";
   final String PATHTODER = "path to .der file";
   final String KEYPAIRID ="CloudFront key pair ID";
   final String HTTPORHTTPS = "https";
   final String CFDISTRIBUTION = "dxxxx.cloudfront.net";
   final String CFPATH = "CloudFront URL for file";
   // date and time that CloudFront's signed URL expires,
   // in Coordinated Universal Time
   final String EXPIRETS = "2012-11-14T22:20:00.000Z";
   final String BLOCKEDCOUNTRY="AU";

   protected void doGet(HttpServletRequest request, HttpServletResponse response)
 throws ServletException, IOException {

      String ip = null;
      StringTokenizer st = null;
      PrintWriter out = response.getWriter();

      String headers = request.getHeader("X-FORWARDED-FOR");

      if (headers!= null){
         st = new StringTokenizer(headers,",");

         while (st.hasMoreTokens()) {
            ip = st.nextToken();
         }
      }

      //Get the client's IP addr in case X-Forwarded-IP header doesn't exist.

      if (ip == null) ip = request.getRemoteAddr();

      try {

         GetMaxMindCountryCode country  = new GetMaxMindCountryCode("GEOAPI
URL","GEOAPITOKEN");

         if ( !country.getCountry(ip).equals(BLOCKEDCOUNTRY)){

            SignedUrl myApp = new SignedUrl(KEYPAIRID,PATHTODER);
          out.println(myApp.getSignedHash(HTTPORHTTPS,CFDISTRIBUTION,CFPATH,EX
PIRETS));

         }else {
            out.println("You cannot access this link.");
         }
      } catch (Exception e1) {
         e1.printStackTrace();
      }
   }
}
```

GetMaxMindCountryCode.java

GetMaxMindCountryCode.java sends MaxMind a request that includes an end user's IP address. The return value is a country code.

```
/*
 * Copyright 2011 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *  http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URL;
import java.net.URLConnection;

public class GetMaxMindCountryCode {

 private static String geoApiEndPoint;
 private static String apiToken;

 GetMaxMindCountryCode(String mygeoApiEndPoint, String myapiToken){
  geoApiEndPoint = mygeoApiEndPoint;
  apiToken = myapiToken;
 }

 public String getCountry(String enduserIP) throws Exception {
  String geoApiURL = "http://"+geoApiEndPoint+"?l="+apiToken+"&i="+enduserIP;

  // Call to MaxMind API.
  URL url = new URL(geoApiURL);
  URLConnection urlConn = url.openConnection();

  urlConn.setUseCaches(false);

  InputStreamReader in = new InputStreamReader((InputStream) urlConn.getCon
tent());
  BufferedReader buff = new BufferedReader(in);

  return buff.readLine();
 }
}
```

SignedUrl.java

SignedUrl.java creates a signed URL that the end user can use to access a file in the CloudFront distribution.

```
/*
 * Copyright 2011 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *  http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.security.Security;
import java.text.ParseException;

import org.jets3t.service.CloudFrontService;
import org.jets3t.service.CloudFrontServiceException;
import org.jets3t.service.utils.ServiceUtils;

public class SignedUrl {
   // Signed URLs for a private distribution
   // Note that Java supports SSL certificates only in DER format,
   // so you need to convert your PEM-formatted file to DER format.
   // To do this, you can use openssl:
   // openssl pkcs8 -topk8 -nocrypt -in origin.pem -inform PEM -out new.der -
outform DER
   // For the encoder to work correctly, you should also add the
   // bouncy castle jar to your project and then add the provider.ds.

   private static String keyPairId;
   private static String privateKeyFilePath;

   SignedUrl(String mykeyPairId, String myprivateKeyFilePath){
      keyPairId = mykeyPairId;
      privateKeyFilePath = myprivateKeyFilePath;
   }

   public String getSignedHash(String protocol, String cfDistribution, String
objectUri, String expTime) throws FileNotFoundException, IOException,
   CloudFrontServiceException, ParseException{

   Security.addProvider(new org.bouncycastle.jce.provider.BouncyCastlePro
vider());

   // Convert your DER file into a byte array.

     byte[] derPrivateKey = ServiceUtils.readInputStreamToBytes(new FileInput
Stream(privateKeyFilePath));

   // Generate a "canned" signed URL to allow access to a
   // specific distribution and object.

   String signedUrlCanned = CloudFrontService.signUrlCanned(
       protocol+ "://" + cfDistribution + "/" + objectUri, // resource URL or
path
       keyPairId,     // Certificate identifier,
                      // an active trusted signer for the distribution
       derPrivateKey, // DER private key data
       ServiceUtils.parseIso8601Date(expTime) // DateLessThan
       );

   return signedUrlCanned;
   }
}
```

PHP Sample Code for MaxMind

The following sample application gets the IP address of the end user and sends the IP address to MaxMind. MaxMind returns the country code that corresponds to the end user's IP address. The application then displays the country code that is blocked and evaluates whether the value returned by MaxMind matches the blocked country code. If the end user's country is not blocked, the application displays a "You are not blocked" message, uses a canned policy to create a signed URL that expires in five minutes, performs the substitutions necessary to ensure that the URL doesn't include any invalid characters, and redirects the user's browser to the signed URL. If the end user's country is blocked, the application displays a "You are blocked" message and a graphic.

```
<!DOCTYPE html>
<html>
<head>
   <title>Geoblocking Test</title>
</head>
<body>
   <h1>Geoblocking Test</h1>

<?php
// Configure the private key (make sure this information is secure).
$private_key_filename = 'path to private key';
$key_pair_id          = 'CloudFront key pair ID';

/*
 * Configure the geoblocking parameters. The following variables
 * describe the two-letter country to be blocked, the
 * CloudFront URL for the file that you want to secure,
 * and the expiry time of the URL. Change these values as needed.
 */
$blocked_geo = 'gb';
$asset_path  = 'CloudFront URL for the object';
$expires     = time() + 300; // (5 minutes from now)

// Configure the URL to the geolocation service.
$token       = 'MaxMind user token';
$address     = 'MaxMind URL';
$remote_ip   = get_remote_ip_address();
$service_url = $address . '?l=' . $token . '&i=' . $remote_ip;

// Call the web service using the configured URL.
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $service_url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
$ws_response = curl_exec($ch);

$edge_geo = $ws_response;

echo '<p>The country being blocked is: ' . strtoupper($blocked_geo) . '</p>';

if ($edge_geo != strtoupper($blocked_geo))
{
   echo '<p>Your country is: ' . strtoupper($edge_geo) . '</p>';
   echo '<p>You are not blocked.</p>';
   $signed_url = create_signed_url($asset_path, $private_key_filename,
$key_pair_id, $expires);
   echo '<img src="' . $signed_url . '" width="600" height="401" />' ;
}
else
{
   echo '<p>Your country is: ' . strtoupper($edge_geo) . '</p>';
   echo '<p>You are blocked.</p>';
  $blocked_url = 'http://s3.amazonaws.com/<Amazon S3 bucket>/blocked-image.jpg';

   echo '<img src="' . $blocked_url . '" alt="Access blocked" width="600"
height="401" />';
}

// Function definitions

function get_remote_ip_address()
{
   // Check to see if an HTTP_X_FORWARDED_FOR header is present.

   if($_SERVER['HTTP_X_FORWARDED_FOR'])

   {

      // If the header is present, use the last IP address.
      $temp_array      = explode(',', $_SERVER['HTTP_X_FORWARDED_FOR']);
      $temp_ip_address = $temp_array[count($temp_array) - 1];
   }
   else
   {
      // If the header is not present, use the
      // default server variable for remote address.
      $temp_ip_address = $_SERVER['REMOTE_ADDR'];
   }

   return $temp_ip_address;
}


function create_signed_url($asset_path, $private_key_filename, $key_pair_id,
$expires)
{
   // Build the policy.
   $canned_policy = '{"Statement":[{"Resource":"' . $asset_path
      . '","Condition":{"DateLessThan":{"AWS:EpochTime":'. $expires . '}}}]}';


   // Sign the policy.
   $signature = rsa_sha1_sign($canned_policy, $private_key_filename);

   // Make the signature contains only characters that
   // can be included in a URL.
   $encoded_signature = url_safe_base64_encode($signature);

   // Combine the above into a properly formed URL name
   $temp_signed_url = $asset_path . '?Expires=' . $expires . '&Signature='
      . $encoded_signature . '&Key-Pair-Id=' . $key_pair_id;

   return $temp_signed_url;
}
```

```
function rsa_sha1_sign($policy, $private_key_filename)
{
   $signature = '';

   // Load the private key.
   $fp = fopen($private_key_filename, 'r');
   $private_key = fread($fp, 8192);
   fclose($fp);

   $private_key_id = openssl_get_privatekey($private_key);

   // Compute the signature.
   openssl_sign($policy, $signature, $private_key_id);

   // Free the key from memory.
   openssl_free_key($private_key_id);

   return $signature;
}

function url_safe_base64_encode($value)
{
   $encoded = base64_encode($value);

   // Replace characters that cannot be included in a URL.
   return str_replace(array('+', '=', '/'), array('-', '_', '~'), $encoded);
}
?>

</body>
</html>
```

.NET Sample Code for MaxMind

The following sample application gets the IP address of the end user and sends the IP address to MaxMind. MaxMind returns the country code that corresponds with the end user's IP address. The application then evaluates whether the value returned by Digital Element matches the blocked country code. If the end user's country is blocked, the application displays a message to that effect. If the end user's country is not blocked, the application creates a signed URL that expires in one minute, performs the substitutions necessary to ensure that the URL doesn't include any invalid characters, and redirects the user's browser to the signed URL.

```
<%@ Page Language="C#" AutoEventWireup="true"  %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "ht
tp://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
   <div>
   <%=GetContent()%>
   </div>
   </form>
</body>
</html>

<%@ Import Namespace="System.Linq" %>
<%@ Import Namespace="System.Xml.Linq" %>
<%@ Import Namespace="System.Security.Cryptography" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>

<script runat="server">

   // Key pair ID for the CloudFront key pair
   private const string KEYPAIR_ID = "CloudFront key pair ID";

   // Private key for the CloudFront key pair.
   // The value is derived from opensslkey.
   private const string PRIVATE_KEY = "private key";

   // JSON policy statement used in the expiring URL
   private const string POLICY = "{{\"Statement\":[{{\"Resource\":\"{0}\",\"Con
dition\":{{\"DateLessThan\":{{\"AWS:EpochTime\":{1}}}}}}}]}}";

   // User token to be passed in to GEO IP service call
   private const string USERTOKEN = "user token";

   // Geolocation service URL with parameters:
   // {0} = User Token and {1} = IP address
   private const string SERVICEURL = "http://geoip3.maxmind.com/a?l={0}&i={1}";


   // Array of countries to block
   private static readonly string[] COUNTRIES_TO_BLOCK = new String[] {"US"};

   private const string BLOCKED_MSG = "Your access to this content is blocked
because you're visiting from '{0}'.";

   /// <summary>
   /// Returns the IP address coming from the request object.
   /// </summary>
   /// <returns>The IP address for the request.</returns>
   private string GetOriginIpAddress()
   {
      // .NET provides Request.UserHostAddress to get the
      // remote IP address, but this could be the IP address of the
      // last proxy in a chain, for example, an Elastic Load Balancer.
      // Instead use the HTTP_X_FORWARDED_FOR header if one exists.
      string forwardedIpAddresses = this.Request.ServerVariables["HTTP_X_FORWAR
DED_FOR"];

      if (string.IsNullOrEmpty(forwardedIpAddresses))
      {
         // Return the UserHostAddress.
         return Request.UserHostAddress;
      }
      else
      {
         // Get the last item in the list.
         return forwardedIpAddresses.Split(',').Last().Trim();
      }
   }

   /// <summary>
   /// This function returns the country code
   /// associated with the IP address in the request object.
   /// </summary>
   /// <returns>The country code for the request.</returns>
   private string GetCountryCodeFromIP()
   {
      var ipAddress = GetOriginIpAddress();
      var serviceURL = String.Format(SERVICEURL, Server.UrlEncode(USERTOKEN),
Server.UrlEncode(ipAddress));

      try
      {
         var webReq = HttpWebRequest.Create(serviceURL);
         var webRes = webReq.GetResponse().GetResponseStream();
         var sr = new StreamReader(webRes);
         var strRes = sr.ReadToEnd();
         sr.Close();
         return strRes.Trim().ToUpper();
      }
      catch(Exception ex)
      {
         // There was an error in making the web request.
         this.Response.Write(serviceURL +  "<br><br>");
         this.Response.Write(ex.Message);
         this.Response.End();
      }
      return null;
   }

   /// <summary>
   /// This function returns a signed URL that will expire
   /// in 1 minute. For more information, see "Create a URL Signature
   /// Using C# and the .NET Framework" in the Amazon CloudFront Developer
Guide:
   /// http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Cre
ateSignatureInCSharp.html
   /// </summary>
   /// <param name="resourceUrl"></param>
   /// <returns></returns>
   private string GetSignedURL(string resourceUrl)
   {
      // Compute expiration date.
      var endTimeSpanFromNow = new TimeSpan(0, 1, 0);
      var intervalEnd = (DateTime.UtcNow.Add(endTimeSpanFromNow)) - new Date
Time(1970, 1, 1, 0, 0, 0, DateTimeKind.Utc);
      var endTimestamp = (int)intervalEnd.TotalSeconds; // Timestamp must be a
 whole number
      var expires = endTimestamp.ToString();
      var strPolicy = string.Format(POLICY, resourceUrl, expires);

      // Encrypt the policy.
      var bufferPolicy = Encoding.ASCII.GetBytes(strPolicy);
      var cryptoSHA1 = new SHA1CryptoServiceProvider();
      bufferPolicy = cryptoSHA1.ComputeHash(bufferPolicy);
      var providerRSA = new RSACryptoServiceProvider();
      providerRSA.FromXmlString(PRIVATE_KEY);
      var rsaFormatter = new RSAPKCS1SignatureFormatter(providerRSA);
      rsaFormatter.SetHashAlgorithm("SHA1");
      var signedPolicyHash = rsaFormatter.CreateSignature(bufferPolicy);
      var strSignedPolicy = System.Convert.ToBase64String(signedPolicyHash);

      // Build the query string with the expiration, policy signature,
      // and CloudFront key pair ID.
      var queryString = "Expires={0}&Signature={1}&Key-Pair-Id={2}";
      queryString = String.Format(queryString, Server.UrlEncode(expires),
Server.UrlEncode(strSignedPolicy), Server.UrlEncode(KEYPAIR_ID));
      var urlString = resourceUrl + "?" + queryString;
      return urlString;
   }

   /// <summary>
   /// Return a message saying this is blocked because of your location,
   /// or return an image tag.
   /// </summary>
   /// <returns></returns>
   public string GetContent()
   {
      var country = GetCountryCodeFromIP();
      if (COUNTRIES_TO_BLOCK.Contains(country))
      {
         // The country returned from the call to the geolocation service
         // is listed in the array of blocked countries.
         return string.Format(BLOCKED_MSG, country);
      }
      else
      {
         // The country returned from the call to the geolocation service
         // is NOT listed in the array of blocked countries
         // Get a signed URL for the content and display it.
         var url = GetSignedURL("CloudFront URL");
         var img = "<img src='{0}' />";
         return String.Format(img, url);
      }
   }
</script>
```

Frequently Asked Questions

How can I ensure that I retrieve the correct IP address of the end user visiting my website?

You can use a variety of methods to get the IP address of the end user visiting your website. Here are two possible methods:

- If your web server is not connected to the Internet through a load balancer, you can use a web server variable to get the remote IP address. However, this IP address isn't always the end user's IP address—it can also be the IP address of a proxy server, depending on how the end user is connected to the Internet.
- If your web server is connected to the Internet through a load balancer, a web server variable may contain the IP address of the load balancer, not the IP address of the end user. In this configuration, we recommend that you use the last IP address in the `X-Forwarded-For` http header. This header

typically contains more than one IP address, most of which are for proxies or load-balancers. The last IP address in the list is the one most likely to be associated with the end user's geographic location.

If your web server is not connected to a load balancer, we recommend that you use web server variables instead of the `X-Forwarded-For` header in order to avoid IP address spoofing. The sample code in this document uses the `X-Forwarded-For` header if the header is present. If you do not want to use this method to get the IP address of the end user, you can edit the sample code.

Can I use any third-party geolocation service to restrict access to my content in CloudFront?

Yes. You will need an account with the third-party service to call their API, and you will need to modify the sample code accordingly.

What is the cost of using this solution?

The cost of using a third-party geolocation service will depend on which service provider you use. Current pricing for CloudFront usage is available on the Amazon CloudFront Pricing page. There are no additional CloudFront charges for using the CloudFront private-content feature.

Can I use location information other than country to block access to my content?

If your geolocation service provides information in addition to the country code, your application can use that information to determine whether you can distribute your content to the end user. Then your application can generate a CloudFront signed URL as described in this tutorial or in Using a Signed URL to Serve Private Content in the Amazon CloudFront Developer Guide.

What should I do if the third-party service is not returning the correct information about an end user?

Confirm that you are correctly calling the API provided by the third-party geolocation service, and that you are using the correct IP address for the end user. If you are still encountering issues with the third-party service or with the accuracy of the data that you receive from the service, contact the service vendor directly.

Additional Services and Documentation

Digital Element Services and Documentation

For information about Digital Element services, see the Digital Element website.

Documentation for Digital Element services is available only with a Digital Element account.

MaxMind Services and Documentation

MaxMind offers a variety of geolocation services and other web services, including the following services:

- MaxMind GeoIP Omni Web Service, http://www.maxmind.com/app/web_services_omni
- MaxMind JavaScript Web Service, http://www.maxmind.com/app/javascript
- Other MaxMind web services, http://www.maxmind.com/app/web_services

The web distribution for each MaxMind API includes documentation and sample programs.

For more information, see Web Services on the MaxMind website.

Amazon Web Services Documentation

- CloudFront, http://aws.amazon.com/documentation/cloudfront
- Amazon S3, http://aws.amazon.com/documentation/s3/

On-Demand Video Streaming Using CloudFront and Adobe Flash Player

When you stream media files using CloudFront, you provide both your media file and the media player with which you want end users to play the media file. To use Adobe Flash Player to stream media files with CloudFront, perform the procedures in the following topics:

1. Creating an Amazon S3 Bucket (p. 269)
2. Creating CloudFront Web and RTMP Distributions (p. 269)
3. Creating a Flash Project Using Adobe Flash Builder (p. 270)
4. Uploading Media and Flash Builder Files to an Amazon S3 Bucket (p. 272)
5. Playing the Media File (p. 273)

This tutorial uses Adobe Flash Builder version 4.6 to generate the files necessary to stream a video using Adobe Flash Player. For more information about Flash Builder, go to the Flash Builder page on the Adobe website. For information about downloading a free trial version of Adobe Flash Builder, go to the Downloads/Adobe Flash Builder 4.6 page.

For a list of the codecs that Flash Player supports, go to the Supported codecs | Flash Player page on the Adobe website.

For more information about streaming media using CloudFront, see Working with RTMP Distributions (p. 60).

Creating an Amazon S3 Bucket

You can upload your media files and your media player files to the same Amazon S3 bucket or to separate buckets. For this tutorial, you'll create one bucket for both a media file and the Flash Player media player files. You'll upload the files later in the process, after you create the Adobe Flash Player files.

To create an Amazon S3 bucket

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In the Amazon S3 console, click **Create Bucket**.
3. In the **Create Bucket** dialog, enter a bucket name.

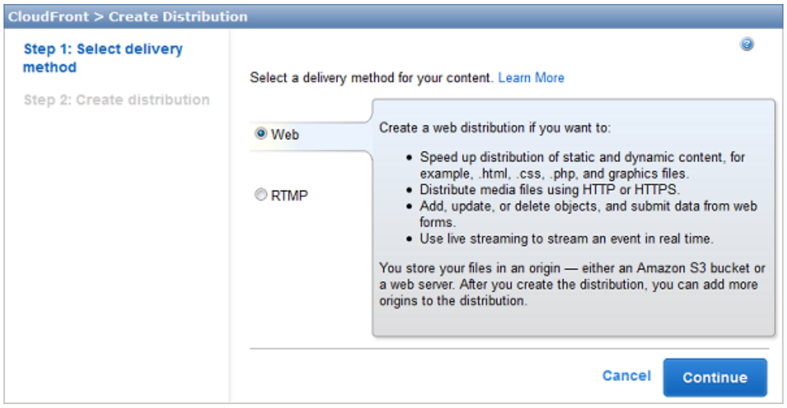
   Important
   For your bucket to work with CloudFront, the name must conform to DNS naming requirements. For more information, go to Bucket Restrictions and Limitations in the *Amazon Simple Storage Service Developer Guide*.
4. Select a region for your bucket. By default, Amazon S3 creates buckets in the US Standard region. We recommend that you choose a region close to you to optimize latency, minimize costs, or to address regulatory requirements.
5. Click **Create**.

Creating CloudFront Web and RTMP Distributions

To configure CloudFront to stream a media file, you need a CloudFront RTMP distribution. For this tutorial, you'll also create a CloudFront web distribution to access the .hlml file that Adobe Flash Builder creates. Perform the following two procedures.

To create a CloudFront web distribution

1. Open the Amazon CloudFront console at https://console.aws.amazon.com/cloudfront/.
2. Click **Create Distribution**.
3. On the first page of the **Create Distribution Wizard**, accept the default selection, **Web**, and click **Continue**.


CloudFront > Create Distribution
Step 1: Select delivery method
Step 2: Create distribution
Web
RTMP
Learn More
Cancel
Continue

Select a delivery method for your content. Learn More

Create a web distribution if you want to:

- Speed up distribution of static and dynamic content, for example, .html, .css, .php, and graphics files.
- Distribute media files using HTTP or HTTPS.
- Add, update, or delete objects, and submit data from web forms.
- Use live streaming to stream an event in real time.

You store your files in an origin — either an Amazon S3 bucket or a web server. After you create the distribution, you can add more origins to the distribution.

4. On the second page of the wizard, click in the **Origin Domain Name** field, and select the Amazon S3 bucket that you created in the procedure To create an Amazon S3 bucket (p. 269). If you have a lot of Amazon S3 buckets, you can type the first few characters of the bucket name to filter the list.
5. Accept the default values for the remaining fields, and click **Create Distribution**.
6. After CloudFront creates your distribution, the value of the **Status** column for your distribution will change from **InProgress** to **Deployed**. This should take less than 15 minutes.

   The domain name that CloudFront assigns to your distribution appears in the list of distributions. (It also appears on the **General** tab for a selected distribution.)

To create a CloudFront RTMP distribution

1. In the CloudFront console, click **Create Distribution**.
2. In the **Create Distribution Wizard**, click **RTMP**, and click **Continue**.
3. On the second page of the wizard, click in the **Origin Domain Name** field, and select the Amazon S3 bucket that you created in the procedure To create an Amazon S3 bucket (p. 269). If you have a lot of Amazon S3 buckets, you can type the first few characters of the bucket name to filter the list.
4. Accept the default values for the remaining fields on the **Create Distribution** page, and click **Create Distribution**.
5. After CloudFront creates your distribution, the value of the **Status** column for your distribution will change from **InProgress** to **Deployed**. This should take less than 15 minutes.

   The domain name that CloudFront assigns to your distribution appears in the list of distributions. The domain name also appears on the **General** tab for a selected distribution.

Creating a Flash Project Using Adobe Flash Builder

You can use Adobe Flash Builder to automatically create a Flash project, which contains all of the files necessary to play a media file using Adobe Flash.

To create a Flash project using Adobe Flash Builder

1. Start Adobe Flash Builder.
2. On the Flash Builder **File** menu, click **New > Flex Project**.
3. Enter the following values:

   - **Project name:** Enter a name for your project, for example, **CloudFrontStreaming**.
   - **Folder:** Specify where you want Flash Builder to save the files for this project. If you don't want to use the default location, uncheck the **Use default location** checkbox, and choose another location.

     Make note of the location; you'll need it later in the process.
   - **Application type:** Accept the default value, **Web**.
   - **Flex SDK version:** Accept the default value, **Use default SDK**.

4. To create the project, click **Finish**.

   After Flash Builder creates the project, a new tab that has the name of your project appears in the Flash Builder user interface. The **Source** button on the **<project-name>** tab is selected, and the **Source** page contains several lines of XML code.
5. Delete the default XML code on the **Source** page.
6. Copy the following XML code, and paste it into the blank **Source** page in Adobe Flash Builder.

```
<?xml version="1.0" encoding="utf-8"?>
   <s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
      xmlns:s="library://ns.adobe.com/flex/spark"
      xmlns:mx="library://ns.adobe.com/flex/mx" minWidth="955" min
Height="600">
   <fx:Declarations>
      <!-- Place non-visual elements here, for example, services and value
 objects -->
   </fx:Declarations>
   <fx:Script>
      <![CDATA[
         import mx.events.FlexEvent;
        import org.osmf.net.StreamingURLResource; import org.osmf.net.FMSURL;

         protected function vp_preinitializeHandler(event:FlexEvent): void
         {
        var myURL:StreamingURLResource = new StreamingURLResource("rtmp://RT
MP-DISTRIBUTION-DOMAIN-NAME/cfx/st/mp4:VIDEO-FILE-NAME-WITHOUT-EXTENSION");

            myURL.urlIncludesFMSApplicationInstance = true;
            myVideoPlayer.source = myURL;
         }
      ]]>
   </fx:Script>
   <s:VideoPlayer id="myVideoPlayer" autoPlay="true" preinitialize="vp_prein
itializeHandler(event)"  x="32" y="52"/>
</s:Application>
```

7. In the XML code that you pasted into the **Source** page, replace the following values:

   - Replace `RTMP-DISTRIBUTION-DOMAIN-NAME` with the CloudFront domain name for your RTMP distribution, for example, `s5c39gqb8ow64r.cloudfront.net`.

- Replace `VIDEO-FILE-NAME-WITHOUT-EXTENSION` with the name of your video file, but exclude the filename extension. For example, if the name of your video is my-vacation.mp4, enter only `my-vacation`.

8. Save your changes.
9. On the Flash Builder **Project** menu, click **Export Release Build**.
10. In the **Export Release Build** dialog box, accept all default values, and click **Finish**.

    Flash Builder creates the files for your project and saves them in the location that you specified in Step 3.

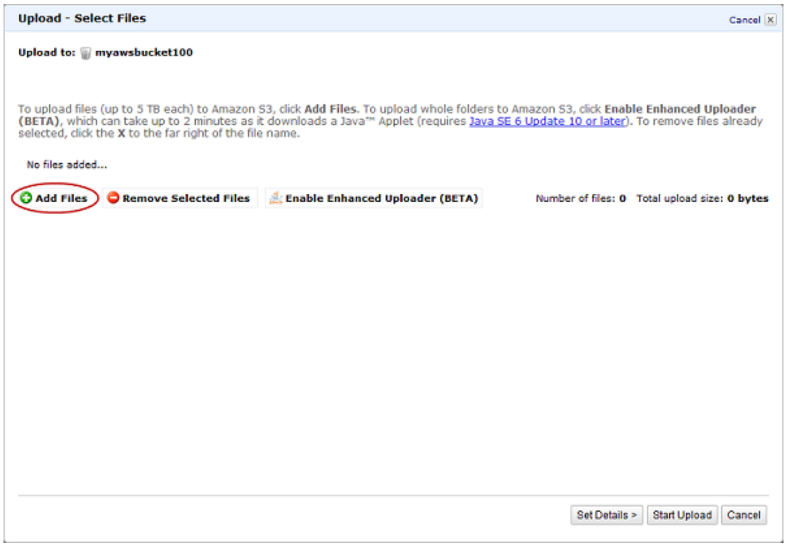
Uploading Media and Flash Builder Files to an Amazon S3 Bucket

When you use Adobe Flash Builder to generate the files for streaming media files, you upload media files and Flash Builder files to the same Amazon S3 bucket.

To upload your media and Flash Builder files to an Amazon S3 bucket

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In the **Buckets** pane, select your bucket, and click **Upload**.
3. On the **Upload - Select Files** page, click **Add Files**, and add the following files:

    - Your media file
    - The files that Flash Builder generated when you performed the procedure To create a Flash project using Adobe Flash Builder (p. 271). Upload only the files in the `bin-release` directory. You can exclude the files in the `bin-release/history` subdirectory.



Upload - Select Files

Upload to: myawsbucket100

To upload files (up to 5 TB each) to Amazon S3, click **Add Files**. To upload whole folders to Amazon S3, click **Enable Enhanced Uploader (BETA)**, which can take up to 2 minutes as it downloads a Java™ Applet (requires Java SE 6 Update 10 or later). To remove files already selected, click the **X** to the far right of the file name.

No files added...

Number of files: **0** Total upload size: **0 bytes**

4. Grant public read permissions for the files that you added in the previous step.

    a. Click **Set Details**.

    b. On the **Set Details** page, click **Set Permissions**.

   c. On the **Set Permissions** page, click **Make everything public**.

5. Click **Start Upload**.

Playing the Media File

To play the media file, you display the HTML file that Flash Builder created for your project and that you uploaded to your Amazon S3 bucket.

To play the media file

1. Enter the CloudFront URL of the HTML file that Flash Builder created for your project by concatenating the following values:

   ```
   http://domain-name-for-your-CloudFront-distribution/HTML-file-name
   ```

   For example:

   ```
   http://d111111abcdef8.cloudfront.net/CloudFrontStreaming.html
   ```
2. In the video player, click the arrow button.

   The video should begin to play.

On-Demand Video Streaming Using CloudFront and Flowplayer for Adobe Flash

When you stream media files using CloudFront, you provide both your media file and the media player with which you want end users to play the media file. To use the Flowplayer for Adobe Flash media player to stream media files with CloudFront, perform the procedures in the following topics:

1. Uploading Media and Flowplayer Files to an Amazon S3 Bucket (p. 274)
2. Creating CloudFront Web and RTMP Distributions (p. 275)
3. Embedding Video in an HTML Page (p. 276)

Note
To stream video using CloudFront and Flowplayer for Adobe Flash, your users must enable Javascript in their browsers.

This tutorial is based on version 3.2.12 of Flowplayer for Adobe Flash. For more information about Flowplayer Flash, go to the Flowplayer Flash website. For a list of the video formats that Flowplayer Flash supports, go to Video formats in the Flowplayer developer documentation about the Flowplayer development environment.

Note
Flowplayer has released an HTML 5 version of their media player. The following procedures only work with Flowplayer Flash, not with Flowplayer HTML5.

For more information about streaming media using CloudFront, see Working with RTMP Distributions (p. 60).

Uploading Media and Flowplayer Files to an Amazon S3 Bucket

You can upload your media files and your media player files to the same Amazon S3 bucket or to separate buckets. For this tutorial, you'll upload an .mp4 media file and the Flowplayer media player files to the same bucket.

To upload media and Flowplayer files to an Amazon S3 bucket

1. Download the following files from the Flowplayer website:

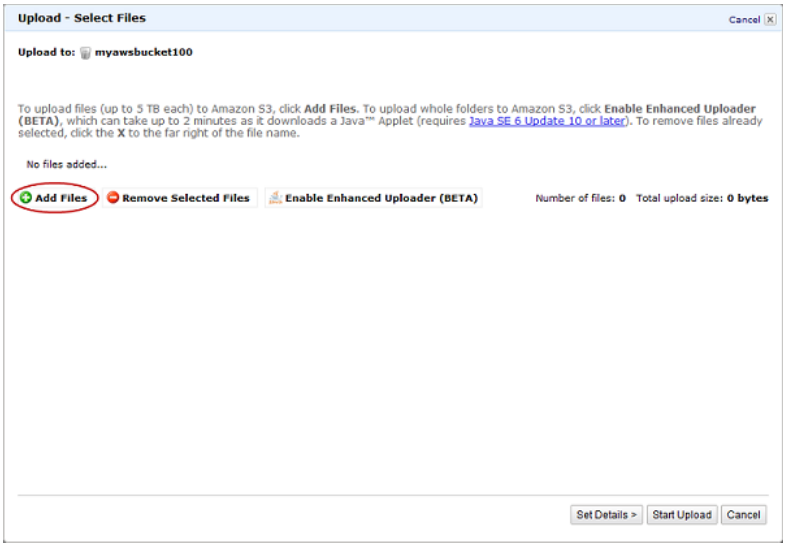
   - The Flowplayer media player. After you download Flowplayer, extract the contents of the .zip file.
   - flowplayer.rtmp-3.2.10.swf. This is a plugin that allows Flowplayer to stream video using the RTMP protocol. The file is available on the RTMP page on the Flowplayer website.

2. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
3. In the Amazon S3 console, click **Create Bucket**.
4. In the **Create Bucket** dialog, enter a bucket name.

   Important
   For your bucket to work with CloudFront, the name must conform to DNS naming requirements. For more information, go to Bucket Restrictions and Limitations in the *Amazon Simple Storage Service Developer Guide*.

5. Select a region for your bucket. By default, Amazon S3 creates buckets in the US Standard region. We recommend that you choose a region close to you to optimize latency, minimize costs, or to address regulatory requirements.
6. Click **Create**.
7. Select your bucket in the **Buckets** pane, and click **Upload**.
8. On the **Upload - Select Files** page, click **Add Files**, and add the following files (the Flowplayer version numbers in your files may be different):

   - flowplayer.controls-3.2.12.swf
   - flowplayer-3.2.11.min.js
   - flowplayer-3.2.12.swf
   - flowplayer.rtmp-3.2.10.swf
   - Your media file in .mp4 format



Upload - Select Files

Upload to: myawsbucket100

To upload files (up to 5 TB each) to Amazon S3, click **Add Files**. To upload whole folders to Amazon S3, click **Enable Enhanced Uploader (BETA)**, which can take up to 2 minutes as it downloads a Java™ Applet (requires Java SE 6 Update 10 or later). To remove files already selected, click the **X** to the far right of the file name.

No files added...

Number of files: 0 Total upload size: 0 bytes

9. Grant public read permissions for the files that you added in the previous step.

   a. Click **Set Details**.
   b. On the **Set Details** page, click **Set Permissions**.
   c. On the **Set Permissions** page, click **Make everything public**.

10. Click **Start Upload**.

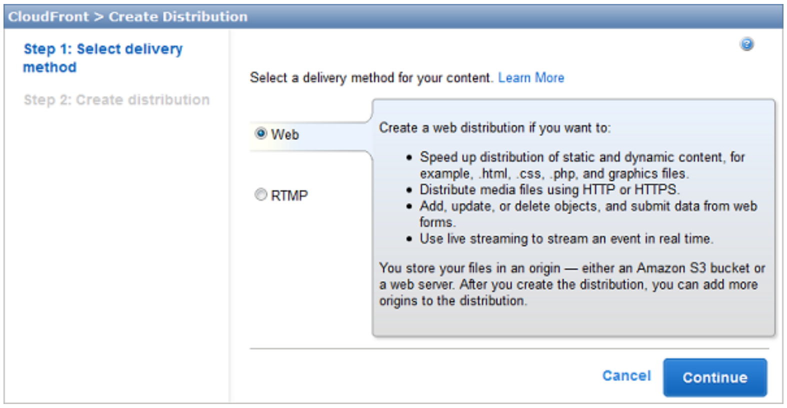
Creating CloudFront Web and RTMP Distributions

To configure CloudFront to stream a media file, you need a CloudFront web distribution for the Flowplayer files and an RTMP distribution for the media file. Perform the following two procedures to create a web distribution and an RTMP distribution.

To create a CloudFront web distribution for your Flowplayer files

1. Open the Amazon CloudFront console at https://console.aws.amazon.com/cloudfront/.
2. Click **Create Distribution**.

3. On the first page of the **Create Distribution Wizard**, accept the default selection, **Web**, and click **Continue**.


CloudFront > Create Distribution
Step 1: Select delivery method
Step 2: Create distribution
● Web
○ RTMP
Cancel
Continue

Select a delivery method for your content. Learn More

Create a web distribution if you want to:

- Speed up distribution of static and dynamic content, for example, .html, .css, .php, and graphics files.
- Distribute media files using HTTP or HTTPS.
- Add, update, or delete objects, and submit data from web forms.
- Use live streaming to stream an event in real time.

You store your files in an origin — either an Amazon S3 bucket or a web server. After you create the distribution, you can add more origins to the distribution.

4. On the second page of the wizard, click in the **Origin Domain Name** field, and select the Amazon S3 bucket that you created in the procedure To upload media and Flowplayer files to an Amazon S3 bucket (p. 274). If you have a lot of Amazon S3 buckets, you can type the first few characters of the bucket name to filter the list.
5. Accept the default values for the remaining fields, and click **Create Distribution**.
6. After CloudFront creates your distribution, the value of the **Status** column for your distribution will change from **InProgress** to **Deployed**. This should take less than 15 minutes.

   The domain name that CloudFront assigns to your distribution appears in the list of distributions. (It also appears on the **General** tab for a selected distribution.)

To create a CloudFront RTMP distribution for your media file

1. In the CloudFront console, click **Create Distribution**.
2. In the **Create Distribution Wizard**, click **RTMP**, and click **Continue**.
3. On the second page of the wizard, click in the **Origin Domain Name** field, and select the Amazon S3 bucket that you created in the procedure To upload media and Flowplayer files to an Amazon S3 bucket (p. 274). If you have a lot of Amazon S3 buckets, you can type the first few characters of the bucket name to filter the list.
4. Accept the default values for the remaining fields on the **Create Distribution** page, and click **Create Distribution**.
5. After CloudFront creates your distribution, the value of the **Status** column for your distribution will change from **InProgress** to **Deployed**. This should take less than 15 minutes.

   The domain name that CloudFront assigns to your distribution appears in the list of distributions. The domain name also appears on the **General** tab for a selected distribution.

Embedding Video in an HTML Page

The following sample HTML file shows you how to stream a video using the web and RTMP distributions that you created in Creating CloudFront Web and RTMP Distributions (p. 275). To use this sample to stream your video, perform the following steps:

1. Copy the HTML code below, and paste it into a text editor.
2. Review the comments in the HTML file, and replace the following placeholders with the applicable values:

- WEB-DISTRIBUTION-DOMAIN-NAME
- VIDEO-FILE-NAME
- RTMP-DISTRIBUTION-DOMAIN-NAME

3. Save the file with a .html filename extension, for example, `flowplayer-example.html`.
4. Open the .html file in a web browser, and play your video.

```
<HTML>
<HEAD>
<TITLE>Amazon CloudFront Streaming with Flowplayer</TITLE>
</HEAD>

<BODY>

<H1>This video is streamed by CloudFront and played in Flowplayer.</H1>

<!-- This HTML file plays an MP4 media file using Flowplayer.

Replace all instances of WEB-DISTRIBUTION-DOMAIN-NAME with the
domain name of your CloudFront web distribution, for example,
d111111abcdef8.cloudfront.net (begins with "d").

Update the version number that appears in the flowplayer-version filenames
with the version number of the files that you downloaded from the Flowplayer
website.
The files may not have the same version number.

Ensure that URLs don't contain any spaces.
-->

<!-- Call the Flowplayer JavaScript file. -->
<script src="http://WEB-DISTRIBUTION-DOMAIN-NAME/flowplayer-
3.2.11.min.js"></script>

<!-- Style section. Specify the attributes of the player
such as height, width, color, and so on.
-->
<style>
a.rtmp {
   display:block;
   width:720px;
   height:480px;
   margin:25px 0;
   text-align:center;
   background-color:black;
}
</style>

<!--  Replace VIDEO-FILE-NAME with the name of your .mp4 video file,
excluding the .mp4 filename extension. For example, if you uploaded a file
called my-vacation-video.mp4, enter my-vacation-video.

If you're streaming an .flv file, use the following format:
<a class="rtmp" href="VIDEO-FILE-NAME"/>
-->
```

```
<a class="rtmp" href="mp4:VIDEO-FILE-NAME"/>

<script type="text/javascript">
$f("a.rtmp", "http://WEB-DISTRIBUTION-DOMAIN-NAME/flowplayer-3.2.12.swf", {
   // Configure Flowplayer to use the RTMP plugin for streaming.
   clip: {
      provider: 'rtmp'
   },

   // Specify the location of the RTMP plugin.
   plugins: {
      rtmp: {
        url: 'http://WEB-DISTRIBUTION-DOMAIN-NAME/flowplayer.rtmp-3.2.10.swf',


         // Replace RTMP-DISTRIBUTION-DOMAIN-NAME with the domain name of your

         // CloudFront RTMP distribution, for example, s5c39gqb8ow64r.cloud
front.net.
         netConnectionUrl: 'rtmp://RTMP-DISTRIBUTION-DOMAIN-NAME/cfx/st'
      }
   }
});
</script>

</BODY>
</HTML>
```

On-Demand Video Streaming Using CloudFront and JW Player

When you stream media files using CloudFront, you provide both your media file and the media player with which you want end users to play the media file. To use the JW Player media player to stream media files with CloudFront, perform the procedures in the following topics:

1. Uploading Media and JW Player Files to an Amazon S3 Bucket (p. 279)
2. Creating CloudFront Web and RTMP Distributions (p. 280)
3. Embedding Video in a Web Page (p. 281)
4. Uploading the HTML File and Playing the Video (p. 282)

This tutorial is based on the free edition of JW Player version 6.1. For more information about JW Player, go to the JW Player website. For a list of the video formats that JW Player supports, go to the JW Player Features page.

For more information about streaming media using CloudFront, see Working with RTMP Distributions (p. 60).

Uploading Media and JW Player Files to an Amazon S3 Bucket

You can upload your media files and your media player files to the same Amazon S3 bucket or to separate buckets. For this tutorial, you'll upload an .mp4 or .flv media file and the JW Player media player files to the same bucket.

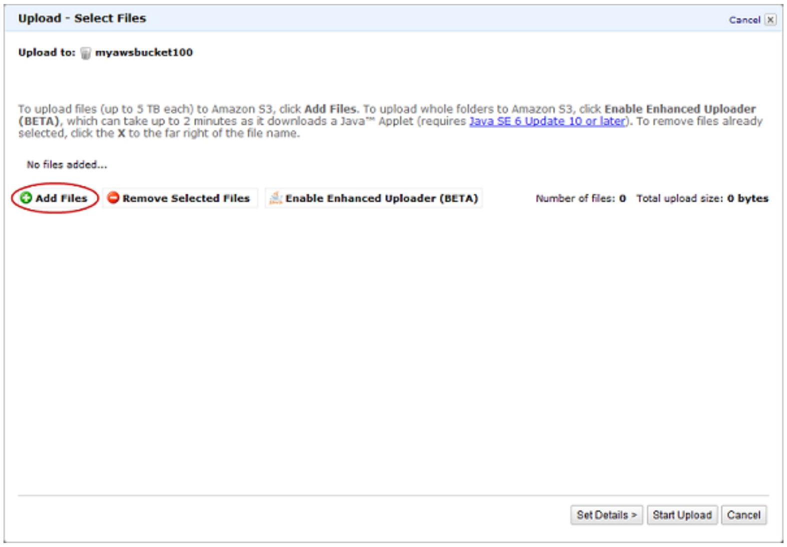
To upload media and JW Player files to an Amazon S3 bucket

1. If you don't already have the files for the JW Player media player, download the player from the Features page on the JW Player website. Then extract the contents of the .zip file.
2. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
3. In the Amazon S3 console, click **Create Bucket**.
4. In the **Create Bucket** dialog, enter a bucket name.

   Important
   For your bucket to work with CloudFront, the name must conform to DNS naming requirements. For more information, go to Bucket Restrictions and Limitations in the *Amazon Simple Storage Service Developer Guide*.

5. Select a region for your bucket. By default, Amazon S3 creates buckets in the US Standard region. We recommend that you choose a region close to you to optimize latency, minimize costs, or to address regulatory requirements.
6. Click **Create**.
7. Select your bucket in the **Buckets** pane, and click **Upload**.
8. On the **Upload - Select Files** page, click **Add Files**, and add the following files:

   - jwplayer.flash.swf
   - jwplayer.html5.js
   - jwplayer.js
   - Your .mp4 or .flv media file.



Upload - Select Files

Upload to: myawsbucket100

To upload files (up to 5 TB each) to Amazon S3, click **Add Files.** To upload whole folders to Amazon S3, click **Enable Enhanced Uploader (BETA)**, which can take up to 2 minutes as it downloads a Java™ Applet (requires Java SE 6 Update 10 or later). To remove files already selected, click the **X** to the far right of the file name.

No files added...

Number of files: **0** Total upload size: **0 bytes**

9. Grant public read permissions for the files that you added in the previous step.

   a. Click **Set Details**.
   b. On the **Set Details** page, click **Set Permissions**.
   c. On the **Set Permissions** page, click **Make everything public**.

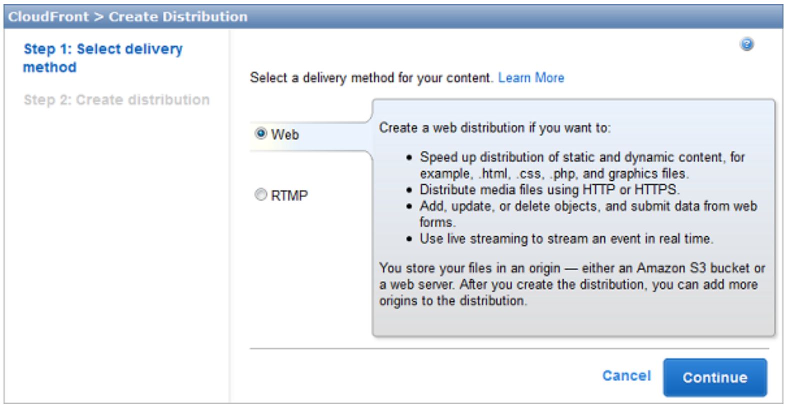
10. Click **Start Upload**.

Creating CloudFront Web and RTMP Distributions

To configure CloudFront to stream a media file, you need a CloudFront web distribution for the JW Player files and an HTML file, and an RTMP distribution for the media file. Perform the following two procedures to create a web distribution and an RTMP distribution.

To create a CloudFront web distribution for your JW Player files

1. Open the Amazon CloudFront console at https://console.aws.amazon.com/cloudfront/.
2. Click **Create Distribution**.
3. On the first page of the **Create Distribution Wizard**, accept the default selection, **Web**, and click **Continue**.



CloudFront > Create Distribution

Step 1: Select delivery method

Step 2: Create distribution

Select a delivery method for your content. Learn More

● Web

○ RTMP

Create a web distribution if you want to:

- Speed up distribution of static and dynamic content, for example, .html, .css, .php, and graphics files.
- Distribute media files using HTTP or HTTPS.
- Add, update, or delete objects, and submit data from web forms.
- Use live streaming to stream an event in real time.

You store your files in an origin — either an Amazon S3 bucket or a web server. After you create the distribution, you can add more origins to the distribution.

4. On the second page of the wizard, click in the **Origin Domain Name** field, and select the Amazon S3 bucket that you created in the procedure To upload media and JW Player files to an Amazon S3 bucket (p. 279). If you have a lot of Amazon S3 buckets, you can type the first few characters of the bucket name to filter the list.
5. Accept the default values for the remaining fields, and click **Create Distribution**.
6. After CloudFront creates your distribution, the value of the **Status** column for your distribution will change from **InProgress** to **Deployed**. This should take less than 15 minutes.

   The domain name that CloudFront assigns to your distribution appears in the list of distributions. The domain name also appears on the Distribution Settings page for a selected distribution.)

To create a CloudFront RTMP distribution for your media file

1. In the CloudFront console, click **Create Distribution**.
2. In the **Create Distribution Wizard**, click **RTMP**, and click **Continue**.
3. On the second page of the wizard, click in the **Origin Domain Name** field, and select the Amazon S3 bucket that you created in the procedure To upload media and JW Player files to an Amazon S3 bucket (p. 279). If you have a lot of Amazon S3 buckets, you can type the first few characters of the bucket name to filter the list.
4. Accept the default values for the remaining fields on the **Create Distribution** page, and click **Create Distribution**.
5. After CloudFront creates your distribution, the value of the **Status** column for your distribution will change from **InProgress** to **Deployed**. This should take less than 15 minutes.

   The domain name that CloudFront assigns to your distribution appears in the list of distributions. The domain name also appears on the Distribution Settings page for a selected distribution.

Embedding Video in a Web Page

The following example shows you how to embed a video in a web page using the web and RTMP distributions that you created in Creating CloudFront Web and RTMP Distributions (p. 280).

> **Note**
> You can also use the JW Player Setup Wizard to get the code that you add to your HTML file. For more information, see the Setup Wizard page on the JW Player website.

Perform the following steps:

1. Copy the HTML code below, and paste it into a text editor.
2. Review the comments in the HTML file, and replace the following placeholders with the applicable values:

   - WEB-DISTRIBUTION-DOMAIN-NAME
   - RTMP-DISTRIBUTION-DOMAIN-NAME
   - VIDEO-FILE-NAME

3. Save the file with a .html filename extension, for example, `jwplayer-example.html`.

```
<HTML>
<HEAD>
<TITLE>Amazon CloudFront Streaming with JW Player 6</TITLE>
```

```
<!-- Call the JW Player JavaScript file, jwplayer.js.
Replace WEB-DISTRIBUTION-DOMAIN-NAME with the domain name of your
CloudFront web distribution, for example, d1234.cloudfront.net
(begins with "d"). This causes a browser to download the JW Player file
before streaming begins.
-->

<script type='text/javascript' src='https://WEB-DISTRIBUTION-DOMAIN-NAME/jwplay
er.js'></script>

</HEAD>

<BODY>
<H1>This video is streamed by CloudFront and played by JW Player 6.</H1>

<!-- Replace RTMP-DISTRIBUTION-DOMAIN-NAME with the domain name of your
RTMP distribution, for example, s5678.cloudfront.net (begins with "s").

Replace VIDEO-FILE-NAME with the name of your .mp4 or .flv video file,
including the .mp4 or .flv filename extension. For example, if you uploaded
my-vacation.mp4, enter my-vacation.mp4.
-->

<div id='mediaplayer'></div>
<script type="text/javascript">
   jwplayer('mediaplayer').setup({
      file: "rtmp://RTMP-DISTRIBUTION-DOMAIN-NAME/cfx/st/VIDEO-FILE-NAME",
      width: "720",
      height: "480"
   });
</script>

</BODY>
</HTML>
```

Uploading the HTML File and Playing the Video

To play the video using the HTML file that you created in Embedding Video in a Web Page (p. 281), upload the file to your Amazon S3 bucket, and use the URL for your CloudFront distribution.

To upload the HTML file and play the video

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. Select your bucket, and click **Upload**.
3. On the **Upload - Select Files** page, click **Add Files**, and add your HTML file.
4. Grant public read permissions for the HTML file that you added in the previous step.

   a. Click **Set Details**.
   b. On the **Set Details** page, click **Set Permissions**.
   c. On the **Set Permissions** page, click **Make everything public**.

5. Click **Start Upload**.
6. To play the video, enter the following URL in a web browser:

   ```
   http://domain name of your CloudFront web distribution/your HTML file name
   ```

Amazon CloudFront Resources

Although fairly simple to use, CloudFront is rich in functionality. The resources listed here can help you learn more about CloudFront.

Topics

- Additional Amazon CloudFront Documentation (p. 283)
- Getting Support (p. 284)
- CloudFront Developer Tools and SDKs (p. 284)
- Using CloudFront Logging (p. 284)
- Additional Tips from the Amazon Web Services Blog (p. 284)
- Invalidating Objects (p. 285)
- Distributing Streaming Media (p. 285)
- Tools for Configuring Private Content (p. 285)
- Using CloudFront with a Content Management System (p. 286)

Additional Amazon CloudFront Documentation

The following related resources can help you as you work with this service.

- Amazon CloudFront API Reference – Gives complete descriptions of the API actions, parameters, and data types, and a list of errors that the service returns.
- Document History (p. 287) – A high-level overview of current and previous releases with specific attention to new features, corrections, known issues, and documentation improvements.
- Technical documentation for the Amazon Simple Storage Service (S3) – A detailed discussion of the Amazon S3 service, including the basics of getting started, an overview of the service, a programming reference, and an API reference.
- Amazon CloudFront product information – The primary web page for information about CloudFront, including features and pricing information.
- Terms of Use – Detailed information about our copyright and trademark; your account, license, and site access; and other topics.

Getting Support

Support for CloudFront is available in a number of forms.

- Discussion forums – A community-based forum for developers to discuss technical questions related to CloudFront.
- AWS Support Center – This site brings together information about your recent support cases and results from AWS Trusted Advisor and health checks, as well as providing links to discussion forums, technical FAQs, the service health dashboard, and information about AWS support plans.
- AWS Premium Support Information – The primary web page for information about AWS Premium Support, a one-on-one, fast-response support channel to help you build and run applications on AWS Infrastructure Services.
- Contact Us – Links for inquiring about your billing or account. For technical questions, use the discussion forums or support links above.

CloudFront Developer Tools and SDKs

See the Developer Tools page for links to developer resources that provide documentation, code samples, release notes, and other information to help you build innovative applications with AWS.

In addition, Amazon Web Services provides software development kits for accessing CloudFront programmatically. The SDK libraries automate a number of common tasks, including cryptographically signing your service requests, retrying requests, and handling error responses.

- AWS SDK for Java – Setup and other documentation
- AWS SDK for .NET – Setup and other documentation
- AWS SDK for PHP – Setup and other documentation
- AWS SDK for Ruby – Setup and other documentation

Using CloudFront Logging

The following AWS blog posts discuss enhancements to CloudFront logging as well as some ways to analyze access logs.

- AWS Blog – Amazon CloudFront Request Logging (for content delivered via HTTP)
- AWS Blog – Amazon CloudFront Now Supports Streaming Access Logs (for content delivered via RTMP)
- AWS Blog – Enhanced CloudFront Logs, Now With Query Strings

Additional Tips from the Amazon Web Services Blog

The AWS Blog has a number of posts to help you use CloudFront:

- Improving website performance – Improving Global Application Performance
- Creating secure connections using HTTPS – Amazon CloudFront: HTTPS Access, Another Edge Location, Price Reduction

- Using custom origins – New Amazon CloudFront Feature: Custom Origins
- Learning more about third-party tools for Amazon CloudFront – CloudFront Management Tool Roundup

Invalidating Objects

In addition to the invalidation methods provided by CloudFront, you can use the following third-party tools to invalidate objects.

> **Note**
> These tools were developed by third-party vendors who are not associated with Amazon Web Services. For information on how to use these tools, please refer to the vendor's documentation or contact the vendor.

- CloudBuddy Personal – http://m1.mycloudbuddy.com/index.html
- CloudBerry Explorer – http://cloudberrylab.com
- Ylastic – http://ylastic.com
- Cyberduck – http://cyberduck.ch
- Bucket Explorer – http://www.bucketexplorer.com
- CloudFront Invalidator – http://www.swook.net/p/cloudfront-invalidator.html
- CDN Planet CloudFront Purge Tool – http://www.cdnplanet.com/tools/cloudfront-purge-tool/

You can also search for code samples on Github, https://github.com. Search for the phrase *CloudFront invalidation*.

Distributing Streaming Media

The following third-party sources provide additional information on distributing streaming media.

- StreamingMedia.com – How To Get Started With Amazon CloudFront Streaming
- Ioncannon.net –
  - iPhone Windowed HTTP Live Streaming Using Amazon S3 and CloudFront Proof of Concept
  - HTTP Live Video Stream Segmenter and Distributor
  - iPhone Windowed HTTP Live Streaming Server
- Flowplayer.org – Bandwidth detection: Make sure you reach your entire audience with good quality
- JW Player – About RTMP Streaming

Tools for Configuring Private Content

In addition to the methods provided by CloudFront, the following third-party tools provide web forms for configuring your distribution for private content. Some of the tools also provide web forms for creating signed URLs.

- **CloudBuddy** – Supports configuring a distribution for private content and supports creating signed URLs.

  For more information about using CloudBuddy for CloudFront private content, go to Configuring CloudFront Distribution and Private Content.

  This tool is based on research at CSS CorpLabs for a .NET implementation of CloudFront private URLs.

- **Bucket Explorer** – Supports configuring a distribution for private content

  For information about using Bucket Explorer for CloudFront private content, go to How to Create a Private Distribution on a Bucket.
- **CloudBerry** – Supports configuring a distribution for private content and supports creating signed URLs.

  For information about using CloudBerry for CloudFront private content, go to How to Configure Private Content for CloudFront Streaming with CloudBerry.

  For information on setting a default root object, see How to set CloudFront Default Object with CloudBerry S3 Explorer.

For more information about private content, see the AWS Blog: New Amazon CloudFront Feature: Private Content.

Using CloudFront with a Content Management System

You can use CloudFront with several popular content management systems. The following links tell you how.

Drupal

- Drupal.org – CloudFront Installation
- DrupalModules.com – CloudFront Drupal Module

Sitecore

- NTT Data Advisory Service – AWS CloudFront Sitecore Integration

WordPress

- om4.com – Using Amazon CloudFront with WordPress and WordPress MU
- WordPress.org – W3 Total Cache
- WordPress.org – Simple Amazon S3 Upload Form
- WordPress.org – OSSDL CDN Off-linker
- WordPress.org – My CDN
- Inquisiter.com – Amazon CloudFront CDN with a WordPress Blog

Document History

The following table describes the important changes to the documentation since the last release of CloudFront.

- **API Version:** 2014-01-31
- **Latest documentation update:** August 20, 2014

Change	Description	Date Changed
New Feature	CloudFront now supports more ciphers for forwarding HTTPS requests to custom origin servers. For more information, see Encryption (p. 108).	August 20, 2014
New Feature	For web distributions, CloudFront lets you choose whether you want CloudFront to forward headers to your origin and to cache separate versions of a specified object based on the header values in viewer requests. This allows you to serve different versions of your content based on the device the user is using, the location of the viewer, the language the viewer is using, and a variety of other criteria. For more information, see Configuring CloudFront to Cache Objects Based on Request Headers (p. 78).	June 26, 2014
New Feature	Amazon CloudFront now works with AWS CloudTrail to capture information about every request that your AWS account (including your IAM users) sends to the CloudFront API. Integrating CloudFront and CloudTrail lets you determine which requests were made to the CloudFront API, the source IP address from which each request was made, who made the request, when it was made, and more. For more information about using CloudFront with CloudTrail, see Using AWS CloudTrail to Capture Requests Sent to the CloudFront API (p. 192).	May 28, 2014

Change	Description	Date Changed
New Feature	With this release, for HTTPS viewer requests that CloudFront forwards to a custom origin, CloudFront validates that one of the domain names in the SSL certificate on your origin server matches the domain name that you specify for **Origin Domain Name**. If the domain names don't match, CloudFront responds to viewer requests with an HTTP status code 502 (bad gateway) instead of the requested objects. To enable this functionality, you must specify an **Origin Protocol Policy** of **Match Viewer**. For more information, see How to Require HTTPS for Communication between Viewers, CloudFront, and Your Origin (p. 169).	May 16, 2014
New Feature	This release of CloudFront introduces a new field in CloudFront access logs for web distributions. The `time-taken` field shows the number of seconds between the time a CloudFront edge server receives a viewer's request and the time that CloudFront writes the last byte of the response to the server's output queue as measured on the server. For more information about the file format of CloudFront access logs for web distributions, see Web Distribution Log File Format (p. 186).	April 28, 2014
Updated Documentation	Live HTTP Streaming Using CloudFront and Adobe Media Server 5.0 (p. 202) has updated procedures for subscribing to Adobe Media Server and for creating an AWS CloudFormation stack.	March 18, 2014
New Feature	This release of CloudFront introduces usage charts that contain a subset of data from the CloudFront usage report. For more information, see CloudFront Usage Charts (p. 11).	March 13, 2014
New Features	This release of CloudFront introduces the following new features: • **Redirect viewer HTTP requests to HTTPS:** You can now configure CloudFront to redirect viewer HTTP requests to HTTPS. For more information, see How to Require HTTPS for Communication between Viewers, CloudFront, and Your Origin (p. 169). • **Server Name Indication (SNI) for HTTPS requests with alternate domain names:** If you're using your domain name in the URLs for your objects, you can now configure CloudFront to serve HTTPS requests to users whose browsers support Server Name Indication (SNI). For more information, see Using Alternate Domain Names and HTTPS (p. 171).	March 5, 2014
New Feature	This release of CloudFront introduces support for HTTP on-demand streaming of media files in the Microsoft Smooth Streaming format. For more information, see Configuring On-Demand Smooth Streaming (p. 57).	February 20, 2014
New Feature	This release of CloudFront introduces support for HTTP 1.1. For more information, see Transfer Encoding (p. 115). In addition, we added documentation about on-demand progressive downloads and on-demand Apple HTTP live streaming. For more information, see Configuring On-Demand Progressive Downloads and Configuring On-Demand Apple HTTP Live Streaming (HLS) in the *Amazon CloudFront Developer Guide*.	February 7, 2014

Change	Description	Date Changed
New Features	This release of CloudFront introduces geo restriction. If you need to prevent users in selected countries from accessing your content, you can configure a CloudFront web distribution to do one of the following: • Allow users to access content only if they're in a whitelist of specified countries. • Prevent users from accessing content if they're in a blacklist of specified countries. For more information, see Restricting the Geographic Distribution of Your Content (p. 56).	December 18, 2013
New Features	This release of CloudFront introduces the following features: • **DELETE, OPTIONS, PATCH, POST, and PUT support:** You can now use the `DELETE`, `OPTIONS`, `PATCH`, `POST`, and `PUT` HTTP methods in requests that you send to CloudFront. For more information, see Allowed HTTP Methods (p. 47). • **Distribution types renamed:** CloudFront download distributions are now known as web distributions, and streaming distributions are now known as RTMP distributions. • **New columns in access logs for web distributions:** Access logs for CloudFront web distributions now include three additional columns for each request: `x-host-header`, `cs-protocol`, and `cs-bytes`. For more information, see Web Distribution Log File Format (p. 186).	October 15, 2013
New Features	This release of CloudFront introduces the following features: • **Custom error pages:** You can now serve error pages with your own branding and content instead of the default HTTP error messages, such as "404, page not found." You can also use custom error pages to serve a static page when your web server is unavailable. For more information, see Customizing Error Responses (p. 92). • **Configurable cache duration for error responses:** Also known as error caching minimum TTL, this feature lets you specify how long you want CloudFront to cache each error at CloudFront edge locations. CloudFront previously cached all error responses for five minutes; now you can specify any duration and thereby control how frequently CloudFront checks with your origin after an error. For more information, see Customizing Error Responses (p. 92).	September 23, 2013
New Feature	You can now include the * wildcard in a CloudFront alternate domain name (CNAME), such as *.example.com. This is useful when you want to route all requests for objects in a domain and its subdomains to a CloudFront distribution. For more information, see Using Alternate Domain Names (CNAMEs) (p. 29).	September 18, 2013

Change	Description	Date Changed
Updated Documentation	Documentation about live streaming with Wowza Media Server 3.6 was added. For more information, see Live HTTP Streaming Using Wowza Streaming Engine 4.0 (p. 235).	September 10, 2013
Updated Documentation	The documentation about live streaming with Adobe Flash Media Server was replaced with documentation about live streaming with Adobe Media Server version 5.0. For more information, see Live HTTP Streaming Using CloudFront and Adobe Media Server 5.0 (p. 202).	July 31, 2013
New Features	This release of CloudFront introduces the following features: • **Authentication with AWS Signature Version 4:** If you are using CloudFront API version 2013-05-12 or later, you must authenticate requests by using AWS Signature version 4. For more information, see Authenticating REST Requests in the *Amazon CloudFront API Reference*. • **SSL for CloudFront alternate domain names:** CloudFront now supports using HTTPS and using your own domain name in the URLs for your objects (for example, `http://www.example.com/image.jpg`). For more information, see Using Alternate Domain Names and HTTPS (p. 171). In addition, a simultaneous release of Amazon Route 53 introduces the following CloudFront–related feature: • **Amazon Route 53 aliases to CloudFront distributions:** Amazon Route 53 now supports creating alias resource record sets that route DNS queries to alternate domain names for CloudFront distributions. You can use this feature both for alternate domain names at the zone apex (example.com) and alternate domain names for subdomains (www.example.com). For more information, see Routing Queries to an Amazon CloudFront Distribution in the *Amazon Route 53 Developer Guide*.	June 11, 2013
New Features	This release of CloudFront introduces the following features: • **Fields for private content in the AWS Management Console:** Settings for private content, which previously could be configured or changed only using the CloudFront API, can now be configured or changed in the AWS Management Console. This includes settings for origin access identities and trusted signers. In addition, the documentation about private content was reorganized and clarified. For more information, see Serving Private Content through CloudFront (p. 118). • **Improvements to the AWS Management Console:** Wizards and dialog boxes in the AWS Management Console have been resized to simplify viewing on tablet computers without compromising the appearance for other viewers. In addition, the number of pages in the Create Distribution wizard was reduced to simplify the process of creating a new distribution.	September 27, 2012

Change	Description	Date Changed
New Features	This release of CloudFront introduces the following features: • **Access log improvements for web distributions:** For web distributions, CloudFront access logs now include fields for: ◦ The cookie header in each viewer request, including name-value pairs and attributes. This field is optional. ◦ The result type of a request (for example, Hit, RefreshHit, or Miss). ◦ An identifier that uniquely identifies each request (the CloudFront request ID). For more information, see Web Distribution Log File Format (p. 186). • **Cookie support for web distributions:** You can now choose whether you want CloudFront to forward cookies and the associated cookie attributes to your origin. If so, you can also choose whether to forward all cookies or just a selected list of cookies. For more information, see Configuring CloudFront to Cache Objects Based on Cookies (p. 76). • **Price classes for web and RTMP distributions:** You can now choose a price class that corresponds with the maximum price that you want to pay for CloudFront service. If you're willing to accept higher latency for your viewers in some geographic regions in return for lower cost, you can choose a price class that doesn't include all CloudFront regions. For more information, see Choosing the Price Class for a CloudFront Distribution (p. 32).	September 5, 2012
New Features	This release of CloudFront introduces the following features: • You can now invalidate objects using the CloudFront console. For more information, see Invalidating Objects (Web Distributions Only) (p. 87). • The CloudFront console was updated to better support viewing on tablet devices.	June 22, 2012

Change	Description	Date Changed
New Features	This release of CloudFront introduces the following features for web distributions: • You can forward query strings to your origin. For more information, see Configuring CloudFront to Cache Based on Query String Parameters (p. 75). • You can specify up to 10 origins. For more information, see Values that You Specify When You Create or Update a Web Distribution (p. 40). • You can specify path patterns. For more information, see Values that You Specify When You Create or Update a Web Distribution (p. 40). In addition, the CloudFront console has been updated. For more information, see Task List for Creating a Web Distribution (p. 36) and Task List for Streaming Media Files Using RTMP (p. 62). The *Amazon CloudFront Getting Started Guide* was merged into the *Amazon CloudFront Developer Guide*, and the *Amazon CloudFront Developer Guide* was reorganized to enhance usability.	May 13, 2012
Updated Documentation	The documentation about working with objects was reorganized and clarified. For the revised documentation, see Working with Objects (p. 72).	April 4, 2012
New Documentation	Documentation about live streaming with Microsoft IIS Media Services version 4.1 was added. For more information, see Live Smooth Streaming Using Amazon CloudFront and IIS Media Services 4.1 (p. 220).	April 1, 2012
Updated Documentation	The documentation about live streaming with Adobe Flash Media Server was updated with information about Adobe Flash Media Server version 4.5. As of July 31, 2013, CloudFront supports live streaming with Adobe Media Server 5.0. For more information, see Live HTTP Streaming Using CloudFront and Adobe Media Server 5.0 (p. 202).	March 29, 2012
New Feature	This release of CloudFront reduces the minimum TTL value for a web distribution. If you don't specify a minimum TTL when you create a distribution, CloudFront sets the minimum TTL to 0 seconds. For more information, go to the following documentation: • CloudFront product page • "Caching Duration and Minimum TTL" at Request and Response Behavior for Amazon S3 Origins (p. 101) • "Caching Duration and Minimum TTL" at Request and Response Behavior for Custom Origins (p. 106) • The `CachingBehavior` element in the DistributionConfig Complex Type.	March 15, 2012

Change	Description	Date Changed
Updated Documentation	Topics about live streaming with Adobe Flash Media Server and about geoblocking were moved from a separate document into the CloudFront Tutorials (p. 202) chapter in this guide.	February 2, 2012
New Feature	This release of CloudFront introduces AWS Management Console support for creating a distribution with a custom origin, restricting your distribution to HTTPS exclusively, and specifying a default root object. For more information, go to the Amazon CloudFront product page or see any of the following topics in the *Amazon CloudFront Developer Guide*: • Task List for Creating a Web Distribution (p. 36) • Using an HTTPS Connection to Access Your Objects (p. 168) • Specifying a Default Root Object (Web Distributions Only) (p. 96)	April 27, 2011
New Feature	This release of CloudFront includes integration with AWS Identity and Access Management (IAM). For more information, go to the Amazon CloudFront product page or Using IAM to Control Access to CloudFront Resources (p. 178) in the *Amazon CloudFront Developer Guide*.	March 10, 2011
New Feature	This release of CloudFront includes new APIs to support custom origins. For more information, go to the Amazon CloudFront product page or Task List for Creating a Web Distribution (p. 36) in the *Amazon CloudFront Developer Guide*.	November 9, 2010
New Feature	This release of CloudFront includes new APIs for object invalidation. For more information, go to the Amazon CloudFront product page or Invalidating Objects (Web Distributions Only) (p. 87) in the *Amazon CloudFront Developer Guide*.	August 31, 2010
New Feature	CloudFront now supports the ability to assign a default root object to your distribution. For more information, see Specifying a Default Root Object (Web Distributions Only) (p. 96).	August 5, 2010
New Feature	Access logging for HTTP distributions now includes a field for query string parameters. For more information, see Web Distribution Log File Format (p. 186).	July 14, 2010
New Feature	Added support for secure connections using HTTPS. For more information, see Using an HTTPS Connection to Access Your Objects (p. 168).	June 7, 2010
New Feature	Added logging for RTMP content. For more information, see RTMP Distribution Log File Format (p. 189).	May 13, 2010
New Feature	Reduced the minimum amount of time an object can be on an edge server from 24 hours to 1 hour. The default, however, remains 24 hours. For more information, see Specifying How Long Objects Stay in a CloudFront Edge Cache (Expiration) (p. 83).	April 13, 2010
New Feature	Added feature to serve private streaming content over a Real-Time Messaging Protocol (RTMP) and prevent the content from being downloaded. For more information, see Serving Private Content through CloudFront (p. 118).	March 28, 2010

Change	Description	Date Changed
New Feature	Added feature to deliver streaming content over a Real-Time Messaging Protocol (RTMP) connection. For more information, see Task List for Streaming Media Files Using RTMP (p. 62).	December 15, 2009
New Feature	Added feature to restrict access to your content delivered over HTTP. For more information, see Serving Private Content through CloudFront (p. 118).	November 11, 2009
New Guide	We've separated the API reference material into its own guide. The *Amazon CloudFront Developer Guide* contains general information about how to use CloudFront, and the Auto Scaling API Reference contains detailed information about the control API requests, responses, and errors.	November 11, 2009

AWS Glossary

For the latest AWS terminology, see the AWS Glossary in the *AWS General Reference*.