Amazon CloudFront

开发人员指南

API Version 2013-11-11

Amazon CloudFront: 开发人员指南

Amazon CloudFront 简介

Topics

什么是 Amazon CloudFront？为什么我需要它？

CloudFront 是一项 Web 服务，可以加速向最终用户分发您的静态和动态 Web 内容，例如，.html、.css、.php 和图像文件。CloudFront 通过一个由遍布全球的数据中心（称作节点）组成的网络来传输您的内容。当用户请求您用 CloudFront 提供的内容时，用户的请求将被传送到延迟（时延）最短的节点，以便以可以达到的最佳性能来传输内容。如果该内容已经在延迟最短的节点上，CloudFront 将直接提供它。如果该内容目前不在这样的节点上，CloudFront 将从您已指定为该内容最终版本来源的 Amazon S3 存储桶或 HTTP 服务器（例如，Web 服务器）检索该内容。

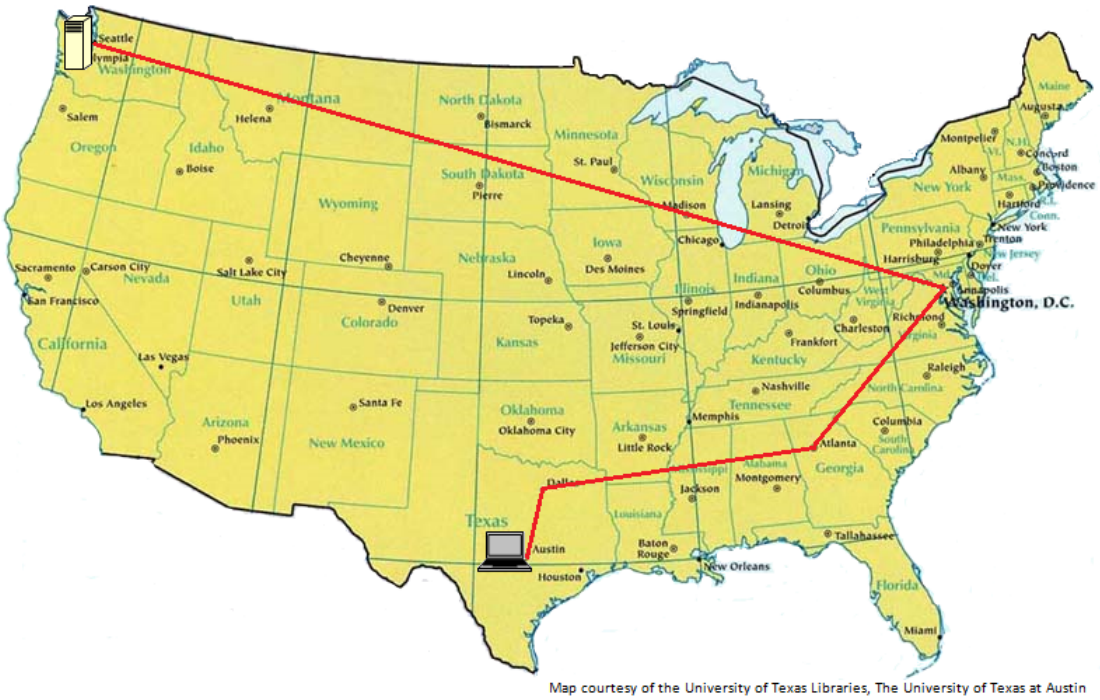
最好通过一个例子来说明这个概念。假设您是从一个传统的 Web 服务器而不是 CloudFront 中提供以下图像：

（该图像归美国宇航局所有，来自可视地球数字图书馆网站：http://visibleearth.nasa.gov/。）

您提供该图像时所用的 URL 为 `http://example.com/globe_west_540.jpg`。您的用户可以轻松导航到该 URL 来查看该图像，但他们可能并不知道他们的请求会不断地在不同网络间传送（通过复杂的互连网络集合传送，这些网络构成了互联网），直到找到该图像为止。

再假设您是从位于美国华盛顿州西雅图的 Web 服务器提供该图像的，并且请求该图像的用户位于美国德克萨斯州的奥斯汀。下面的追踪路由列表（www.WatchMouse.com 友情提供）显示了该请求可能采用的一种传送方式。

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

在该示例中，该请求在美国境内历经 10 次传送后才检索到相应的图像，这一跃点数目也算不上特别大。如果您的用户在欧洲，该请求可能会经由更多网络进行传送，才能到达您在西雅图的服务器。该请求及相应图像必须经由的网络数量和传送的距离对图像的性能、可靠性和可用性有重大影响。

CloudFront 将每个用户请求传送到能以最佳方式提供您的内容的节点，以此来加速分发您的内容。通常，这种节点是延迟最短的 CloudFront 节点。这大大降低了您用户的请求必须经由的网络数量，从而可提高性能。用户将会体验到延迟（加载对象第一个字节所花费的时间）更短、数据传输速率更高。您还会获得更高的可靠性和可用性，因为您的对象的副本现在存储在全球各地的多个节点上。

有关 CloudFront 节点服务器位置的列表，请参阅 Amazon CloudFront 节点网络中的 CloudFront 详细信息页面。

CloudFront 如何传输内容

经过一些初始设置后，CloudFront 可以在无形之中加速传输您的内容。本概述包括在您的首位用户访问您的应用程序或网站前您需要执行的步骤，以及配置完毕后 CloudFront 如何提供您的内容。

CloudFront 的设置过程包括几个简单的步骤：

如何配置 CloudFront 以便传输您的内容

1. 配置您的原始服务器，CloudFront 将从这些服务器中获取您的文件，以便从遍布全球的 CloudFront 节点进行分发。

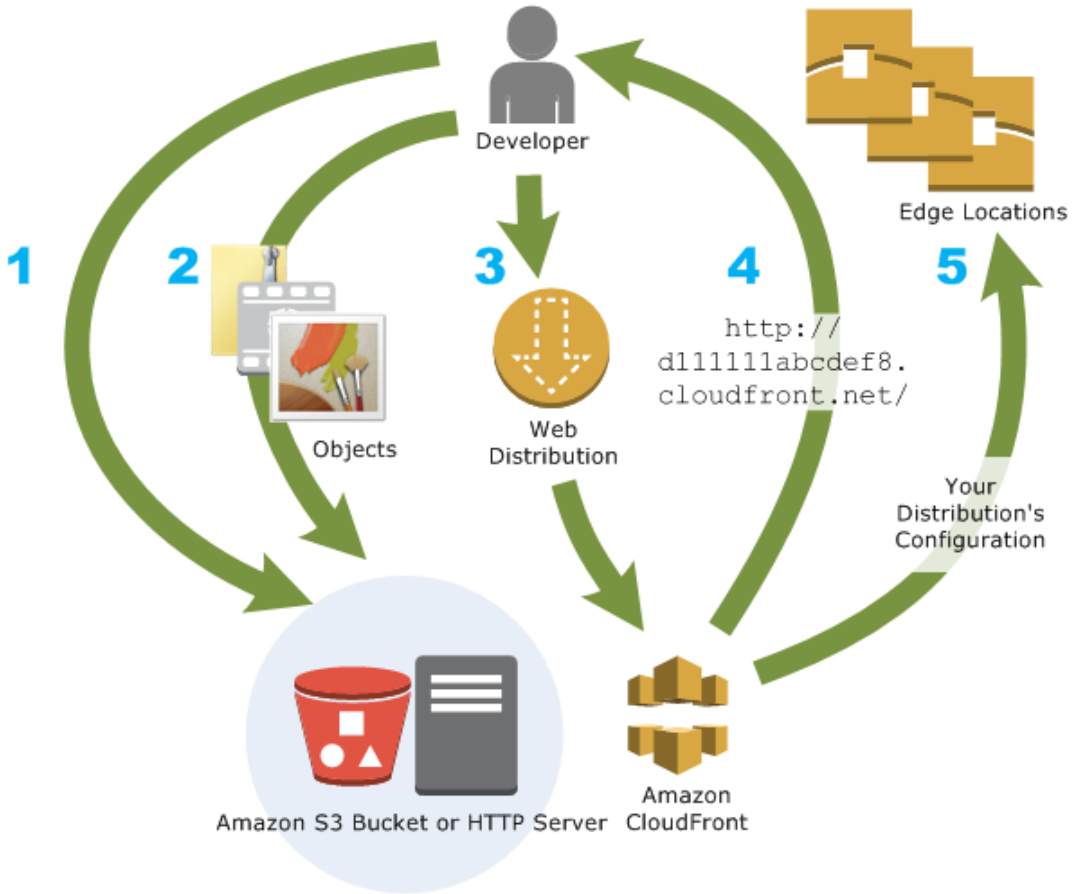
   原始服务器用来存储您的对象的原始最终版本。如果您通过 HTTP 提供内容，您的原始服务器将为 Amazon S3 存储桶或 HTTP 服务器，例如，Web 服务器。您的 HTTP 服务器可在 Amazon Elastic Compute Cloud (Amazon EC2) 实例或您管理的服务器上运行；这些服务器也被称为“自定义源”。

   如果您是使用 Adobe Media Server RTMP 协议按需分发媒体文件，那么您的原始服务器始终为 Amazon S3 存储桶。

2. 将您的文件上传至您的原始服务器。您的文件也称作对象，通常包括网页、图像和媒体文件，但可以是可通过 HTTP 或受支持的 Adobe RTMP（Adobe Flash Media Server 使用的协议）版本提供的任何内容。

   如果您是将 Amazon S3 存储桶用作原始服务器，您可以将存储桶中的对象设为公开可读，这样知道这些对象的 CloudFront URL 的任何人都可以访问它们。您还可以选择将对象设为私有，并控制哪些人可以访问它们。请参阅 通过 CloudFront 提供私有内容。 (p. 91)。
3. 创建一项 CloudFront 分配，此项分配将在用户通过您的网站或应用程序请求文件时告诉 CloudFront 从哪些原始服务器获取您的文件。同时，您还需指定一些详细信息，如您是否希望 CloudFront 记录所有请求以及您是否希望此项分配创建后便立即启用。
4. CloudFront 将此项分配的配置（而不是您的内容）发送到其所有节点。这些节点即服务器的集合，位于分散在不同地理位置的数据中心内；您对象的副本就是被 CloudFront 缓存在这些数据中心。
5. 您在开发网站或应用程序时，需使用 CloudFront 为您的 URL 提供的域名。例如，如果 CloudFront 返回 `d111111abcdef8.cloudfront.net` 作为此项分配的域名，则 Amazon S3 存储桶中（或 HTTP 服务器上的根目录中）logo.jpg 的 URL 将为 `http://d111111abcdef8.cloudfront.net/logo.jpg`。

   您也可以将此项 CloudFront 分配配置为允许使用您自己的域名。在这种情况下，URL 可能是 `http://www.example.com/logo.jpg`。
6. 您还可以选择将您的原始服务器配置为向文件添加标头；标头用于指示您希望文件在 CloudFront 节点上的缓存中保留的时长。默认情况下，每个对象在节点中保留 24 个小时后即会过期。最短过期时间为 0 秒；过期时间无上限。有关更多信息，请参阅 指定对象在 CloudFront 边缘缓存中的保留时间（过期）(p. 63)。



CloudFront 如何将内容传输给您的用户

配置 CloudFront 以传输您的内容后，用户请求您的对象时将发生以下操作：

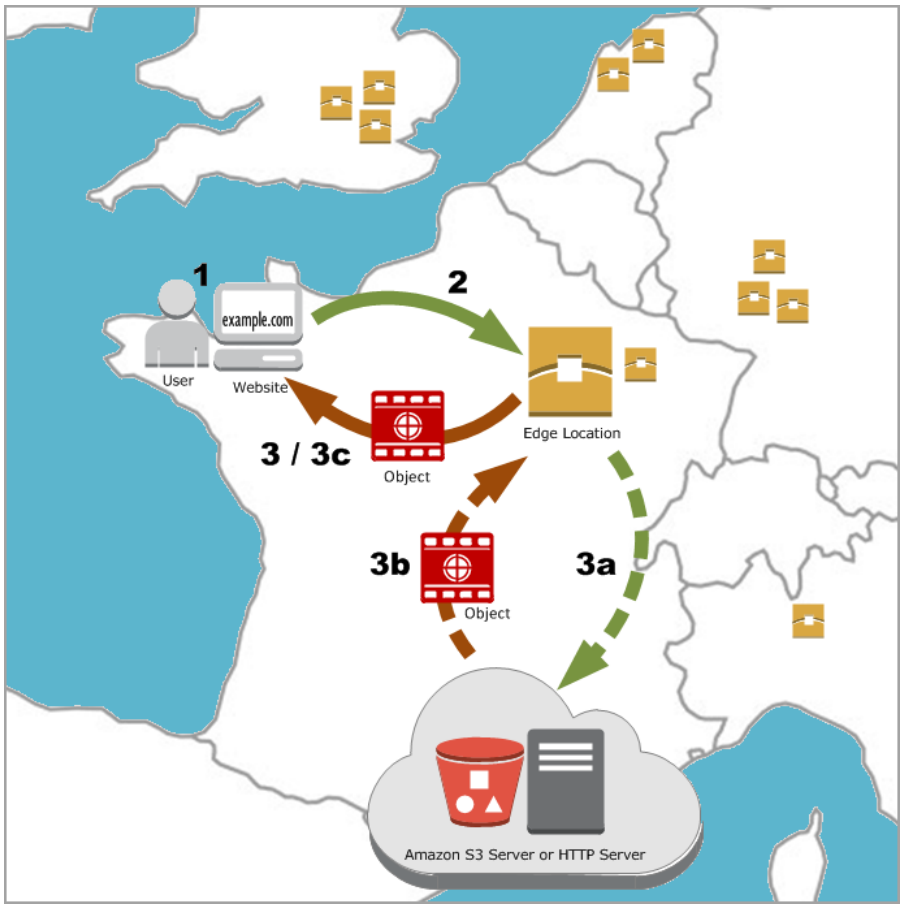
1. 用户访问您的网站或应用程序，并请求一个或多个对象，例如图像文件和 HTML 文件。
2. DNS 将该请求传送到能以最佳方式满足该用户请求的 CloudFront 节点，通常会考虑延迟因素，选择最近的 CloudFront 节点，然后将请求传送到该节点。
3. 在该节点中，CloudFront 会检查其缓存中是否存在所请求的文件。如果这些文件在缓存中，CloudFront 则将它们发回给用户。如果这些文件*不*在缓存中，它会执行以下操作：

   a. CloudFront 将比较该请求与您所创建的分配中的具体说明，然后根据对应的文件类型将文件请求转发到适用的原始服务器：例如，对于图像文件，转发到 Amazon S3 存储桶，对于 HTML 文件，则转发到 HTTP 服务器。
   b. 原始服务器将这些文件发回 CloudFront 节点。
   c. 来自源的第一个字节送达后，CloudFront 便开始将这些文件转发给用户。CloudFront 还会将这些文件添加到节点中的缓存，以便下次有人请求这些文件时使用。

4. 对象在节点缓存中保存 24 个小时后，或者保存时间超过您在文件标头中指定的持续时间后，CloudFront 将执行以下操作：

   a. CloudFront 将下一个针对该对象的请求转发到源，以确定节点中保存的是否为最新版本。
   b. 如果节点中的版本是最新的，CloudFront 则将其传送给您的用户。

      如果节点中的版本不是最新的，您的源则将最新版本传发送给 CloudFront，然后 CloudFront 将对象传送给您的用户并将最新版本存储在该节点的缓存中。



CloudFront 节点服务器的位置和 IP 地址范围

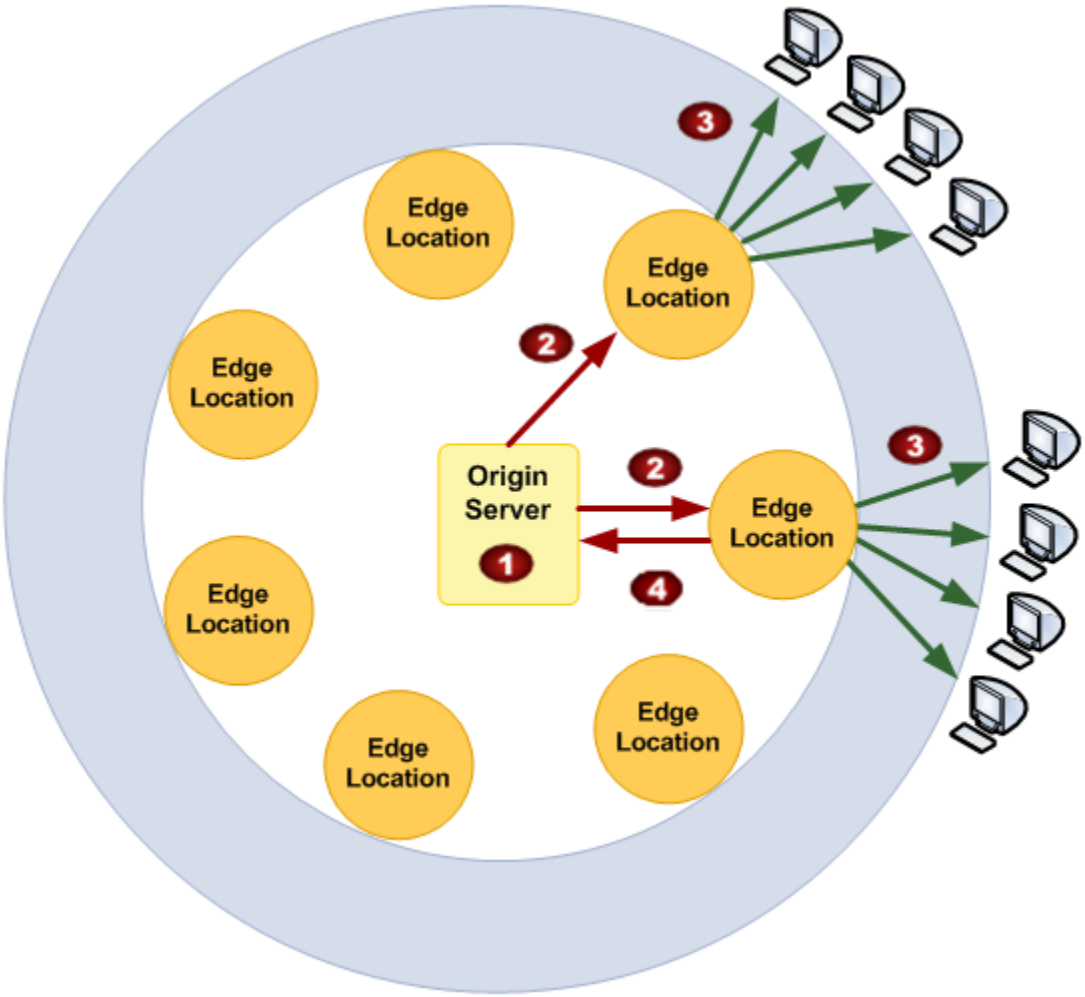
有关 CloudFront 节点服务器位置的列表，请参阅 Amazon CloudFront 节点网络中的 Amazon CloudFront 详细信息页面。

有关 CloudFront 节点服务器的 IP 地址范围列表，请参阅 Amazon CloudFront 开发论坛上的 Amazon CloudFront 公有 IP 范围。

CloudFront 计费和使用

CloudFront 的设计使您不必支付任何前期费用或承诺您会使用多少内容。与使用其他 AWS 服务一样，您按需付费，只为所使用的内容付费。

下图和下表概括了使用 CloudFront 的收费情况。



AWS 为您提供的月账单按 AWS 服务和功能将您的使用情况和具体费用金额划分开来。这样，您就会看到：使用 Amazon S3 来存储对象的一些费用 (1)（如果您是使用 Amazon S3 作为您的原始服务器）；在您的存储桶和您的节点之间传输数据的一些费用 (2)；以及从 CloudFront 提供数据的一些费用 (3)。

	收费项目	注释
1	在 Amazon S3 原始服务器中存储内容	您支付常规的 Amazon S3 存储费用，以将对象存储在您的存储桶中；这类费用显示在您 AWS 对账单的 Amazon S3 部分中。
2	将对象复制到节点	如果您正在使用 Amazon S3 原始服务器，您会产生与 GET 请求和数据传出相对应的常规 Amazon S3 费用。CloudFront 仅在节点需要某个对象时才会将该对象复制到该节点。 数据传输费用显示在您 AWS 对账单的“AWS Data Transfer (AWS 数据传输)”部分。
3	从节点提供对象	您产生与请求和数据传出相对应的 CloudFront 费用，这类费用要低于相应的 Amazon S3 费用。这类 CloudFront 费用显示在您 AWS 对账单的 CloudFront 部分。有关更多信息，请参阅 Amazon CloudFront 定价。

	收费项目	注释
4	将数据提交到源	当用户向您的源传输数据（包括 DELETE、OPTIONS、PATCH、POST 和 PUT 请求）时，您将产生 CloudFront 费用。这类 CloudFront 费用显示在您 AWS 对账单的 CloudFront 部分。有关更多信息，请参阅 Amazon CloudFront 定价。

Note

您还会产生与 HTTPS 请求相对应的额外费用。有关更多信息，请参阅 Amazon CloudFront 定价。

解释您的 AWS 账单和 CloudFront 使用率报告

AWS 向您发送的 CloudFront 服务账单包含可能并非一目了然的代码和缩写。下表中的第一列列出了您的账单中显示的项目，并解释了每个项目的具体含义。

此外，您也可以获取 AWS 提供的 CloudFront 使用率报告，该报告包含的详细信息多于 AWS 提供的 CloudFront 账单。该表中的第二列列出了使用率报告中显示的项目，并显示了账单项目与使用率报告项目之间的关联。

两列中的大多数代码都包含一个由两个字母组成的缩写，该缩写指示了活动的发生位置。在下表中，*region* 在您的 AWS 账单和使用率报告中将被下面其中一个由两个字母组成的缩写取代：

- AP：中国香港特别行政区、韩国和新加坡（亚太地区）
- AU：澳大利亚
- EU：欧洲
- IN：印度
- JP：日本
- SA：南美洲
- US：美国

有关各区域定价的更多信息，请参阅 Amazon CloudFront 定价。

Note

该表未包含将您的对象从 Amazon S3 存储桶传输到 CloudFront 节点所产生的费用。如果有这类费用，它们将显示在您 AWS 账单中的 AWS Data Transfer (AWS 数据传输) 部分中。

您 CloudFront 账单中的项目	CloudFront 使用率报告中“Usage Type (使用类型)”列中的值
region-DataTransfer-Out-Bytes CloudFront 为 Web 和 RTMP 分配提供的字节总和： • Web 分配：为响应用户的 GET 和 HEAD 请求而从位于 *region* 的 CloudFront 节点提供的总字节数 • RTMP 分配：从位于 *region* 的 CloudFront 节点向最终用户传输的总字节数	Web 分配： • *region*-Out-Bytes-HTTP-Static：通过 HTTP 为 TTL >= 3600 秒的对象提供的字节数 • *region*-Out-Bytes-HTTPS-Static：通过 HTTPS 为 TTL >= 3600 秒的对象提供的字节数 • *region*-Out-Bytes-HTTP-Dynamic：通过 HTTP 为 TTL < 3600 秒的对象提供的字节数 • *region*-Out-Bytes-HTTPS-Dynamic：通过 HTTPS 为 TTL < 3600 秒的对象提供的字节数 RTMP 分配： • *region*-FMS-Out-Bytes
region-DataTransfer-Out-OBytes 仅限 Web 分配：为响应 DELETE、OPTIONS、PATCH、POST 和 PUT 请求而从 CloudFront 节点向您的源传输的总字节数。	*region*-Out-OBytes-HTTP-Proxy 为响应 DELETE、OPTIONS、PATCH、POST 和 PUT 请求而通过 HTTP 从 CloudFront 节点向您的源传输的总字节数。 *region*-Out-OBytes-HTTPS-Proxy 为响应 DELETE、OPTIONS、PATCH、POST 和 PUT 请求而通过 HTTPS 从 CloudFront 节点向您的源传输的总字节数。
region-Requests-Tier1 仅限 Web 分配：HTTP GET 和 HEAD 请求的数目	*region*-Requests-HTTP-Static 为 TTL >= 3600 秒的对象提供的 HTTP GET 和 HEAD 请求的数目 *region*-Requests-HTTP-Dynamic 为 TTL < 3600 秒的对象提供的 HTTP GET 和 HEAD 请求的数目
region-Requests-Tier2-HTTPS 仅限 Web 分配：HTTPS GET 和 HEAD 请求的数目	*region*-Requests-HTTPS-Static 为 TTL >= 3600 秒的对象提供的 HTTPS GET 和 HEAD 请求的数目 *region*-Requests-HTTPS-Dynamic 为 TTL < 3600 秒的对象提供的 HTTPS GET 和 HEAD 请求的数目
region-Requests-HTTP-Proxy 仅限 Web 分配：CloudFront 向您的源转发的 HTTP DELETE、OPTIONS、PATCH、POST 和 PUT 请求的数目	*region*-Requests-HTTP-Proxy 与 CloudFront 账单中的对应项目相同

您 CloudFront 账单中的项目	CloudFront 使用率报告中“Usage Type (使用类型)”列中的值
region-Requests-HTTPS-Proxy 仅限 Web 分配：CloudFront 向您的源转发的 HTTPS DELETE、OPTIONS、PATCH、POST 和 PUT 请求的数目	*region*-Requests-HTTPS-Proxy 与 CloudFront 账单中的对应项目相同
失效 仅限 Web 分配：使对象失效（将对象从 CloudFront 节点中删除）所产生的费用；有关更多信息，请参阅支付对象失效费用 (p. 70)	失效 与 CloudFront 账单中的对应项目相同
SSL-Cert-Custom 仅限 Web 分配：使用 SSL 证书及 CloudFront 备用域名（例如 example.com）而不是使用 CloudFront 为您的分配指定的默认 CloudFront SSL 证书及域名所产生的费用	SSL-Cert-Custom 与 CloudFront 账单中的对应项目相同

一般使用数据

AWS 针对 CloudFront 提供两种使用情况报告：账户活动报告从宏观层面显示了您所用的 AWS 服务的所有活动；使用率报告则概括了特定服务按小时、天或月汇总的活动情况。

AWS 账户活动

您可以在“Account Activity (账户活动)”页面上查看按服务列出的 AWS 使用和收费情况摘要。

查看账户活动摘要

1. 访问 http://aws.amazon.com，然后在标题栏中单击 My Account/Console (我的账户/控制台)。
2. 系统提示登录时请登录。
3. 单击 Account Activity (账户活动)。

使用率报告

AWS 针对 CloudFront 提供了一种使用率报告，该报告的详细程度介于账户活动报告中的概览与 CloudFront 访问日志中的详细信息之间。该使用率报告提供了按小时、天或月汇总的使用数据，并按区域和使用类型（例如传出澳大利亚区域的数据）列出了操作。

如果您是使用 Amazon S3 作为 CloudFront 的源，您可能还需要参考 Amazon S3 的使用率报告，此报告也可通过以下步骤生成。

Note

如需获得 CloudFront 收到的针对您对象的每一个请求的详细信息，您可以打开您的分配的 CloudFront 访问日志。有关更多信息，请参阅 访问日志 (p. 145)。

获得使用率报告

1. 访问 http://aws.amazon.com，然后在标题栏中单击 My Account/Console (我的账户/控制台)。

2. 系统提示登录时请登录。
3. 单击 Usage Reports (使用率报告)。
4. 在 Service (服务) 列表中，单击 Amazon CloudFront。

   如果您使用 Amazon S3 作为 CloudFront 分配的源，您可能还需要为 Amazon Simple Storage Service 运行该报告。
5. 指定您要纳入该报告中的使用情况，然后选择您要用来下载或查看该报告的格式：XML 或 CSV。
6. 按照屏幕上的提示查看或保存该报告。

Note

CloudFront 使用率报告中 `Resource` 列的值是您的分配的 ID。

CloudFront 入门

本主题中的示例让您快速了解如何使用 CloudFront，以：

- 在一个 Amazon Simple Storage Service (Amazon S3) 存储桶中存储对象的原始版本。
- 分配下载内容，如文本或图形。
- 使对象可供所有人访问。
- 在对象 URL 中使用 CloudFront 域名（例如，`http://d111111abcdef8.cloudfront.net/image.jpg`），而不是您自己的域名（例如，`http://www.example.com/image.jpg`）。
- 在 CloudFront 节点中将对象保留默认的 24 小时持续时间。（最短持续时间为 0 秒。）

有关在希望使用其他选项时如何使用 CloudFront 的信息，请参阅创建 Web 分配 (p. 19)或创建 RTMP 分配 (p. 22)。

您仅需执行几个基本的步骤即可开始使用 CloudFront 提供内容。第一步是注册。之后创建 CloudFront 分配，然后使用 CloudFront 域名来引用网页或应用程序中的内容。

第 1 步：注册 Amazon Web Services

如果您尚未注册，请在 http://aws.amazon.com 上注册 Amazon Web Services。只要单击 Sign Up Now (立即注册)，然后输入任何必要的信息。

第 2 步：将您的内容上传到 Amazon S3，然后授予对象权限

Amazon S3 存储桶是一个可以包含对象或文件夹的容器。通过将 Amazon S3 存储桶用作源，CloudFront 几乎可为您分配任何类型的对象，例如，文本、图像和视频。您可创建多个存储桶，并且，您可在 Amazon S3 上存储的数据量没有限制。

默认情况下，您的 Amazon S3 存储桶和其中的所有对象都是私有的 — 只有创建存储桶的 AWS 账户有权读写其中的对象。如果您想允许任何人使用 CloudFront URL 访问 Amazon S3 存储桶中的对象，您必须授予对象的公共读取权限。（这是使用 CloudFront 和 Amazon S3 时最常见的错误之一。您必须明确授予对 Amazon S3 存储桶中每个对象的权限。）

Note

如果您想限制能够下载您的内容的人，可使用 CloudFront 私有内容功能。有关分配私有内容的更多信息，请参阅 通过 CloudFront 提供私有内容。 (p. 91)。

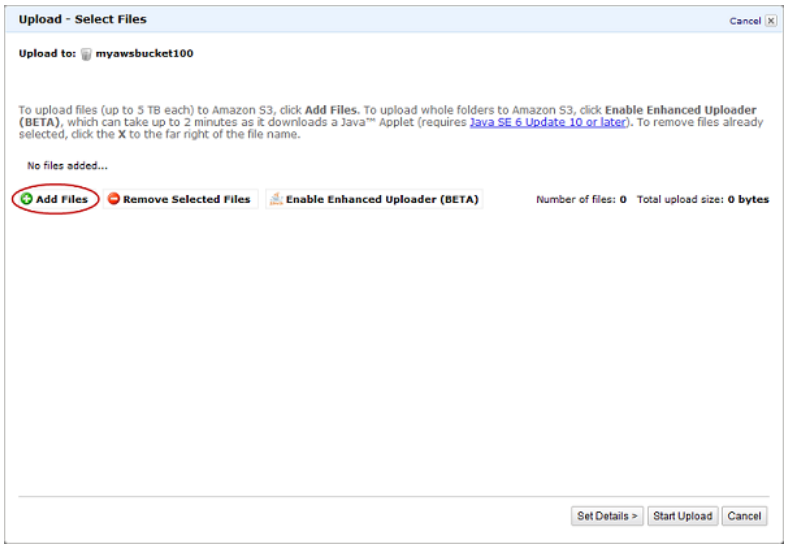
将您的内容上传至 Amazon S3 并授予每个人读取权限

1. 登录 AWS 管理控制台，并通过以下网址打开 Amazon S3 控制台：https://console.aws.amazon.com/s3/。
2. 在 Amazon S3 控制台中，单击 Create Bucket (创建存储桶)。
3. 在 Create Bucket (创建存储桶) 对话框中，输入存储桶名称。

Important

要使您的存储桶能够与 CloudFront 一起使用，其名称必须符合 DNS 命名要求。有关详细信息，请参阅 *Amazon Simple Storage Service 开发者指南* 中的 Bucket Restrictions and Limitations。

4. 选择您的存储桶区域。默认情况下，Amazon S3 在美国标准地区中创建存储桶。我们建议您选择一个靠近您的地区，以便优化延迟，尽可能降低成本或满足法规要求。
5. 单击 Create (创建)。
6. 在 Buckets (存储桶) 窗格中选择您的存储桶，然后单击 Upload (上传)。
7. 在 Upload - Select Files (上传 - 选择文件) 页面上，单击 Add Files (添加文件)，然后选择您要上传的文件。



8. 启用您上传到 Amazon S3 存储桶中的每个对象的公共读取权限。
   a. 单击 Set Details (设置详细信息)。
   b. 在 Set Details (设置详细信息) 页面上，单击 Set Permissions (设置权限)。
   c. 在 Set Permissions (设置权限) 页面上，单击 Make everything public (公开一切信息)。
9. 单击 Start Upload (开始上传)。

上传完成后，您可通过其 URL 导航到该项目。在前面的示例中，此 URL 将是：

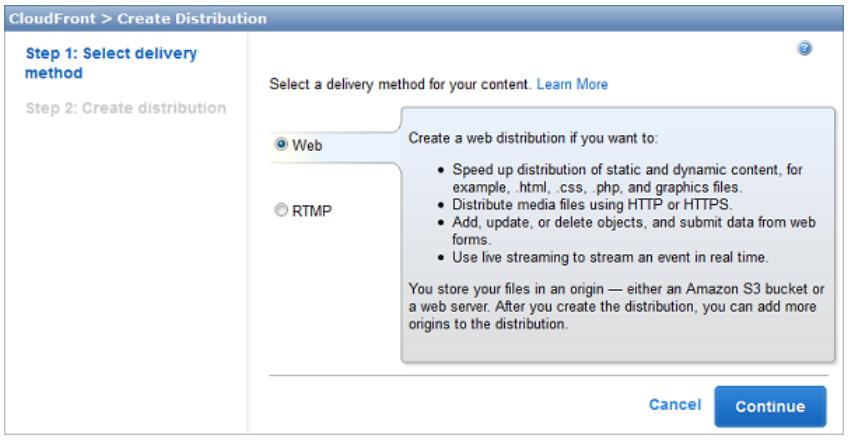
```
http://s3.amazonaws.com/example-myawsbucket/filename
```

使用您的 Amazon S3 URL 来验证您的内容是否可公开访问，但记住，这不是您准备分配内容时将使用的 URL。

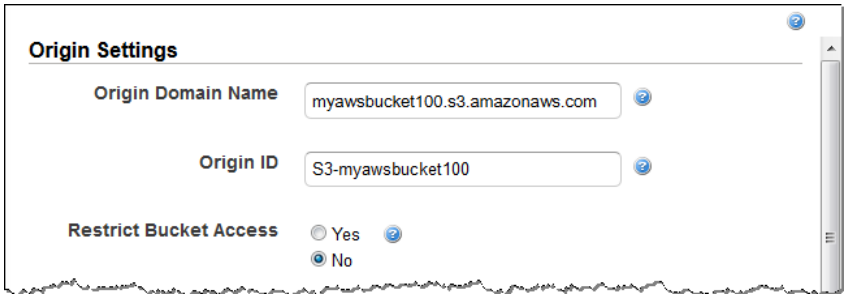
步骤 3：创建 CloudFront Web 分配

创建 CloudFront Web 分配

1. 通过以下网址打开 Amazon CloudFront 控制台：https://console.aws.amazon.com/cloudfront/。
2. 单击 Create Distribution (创建分配)。
3. 在 Create Distribution Wizard (创建分配向导) 的第一页上，接受默认选择 Web，然后单击 Continue (继续)。



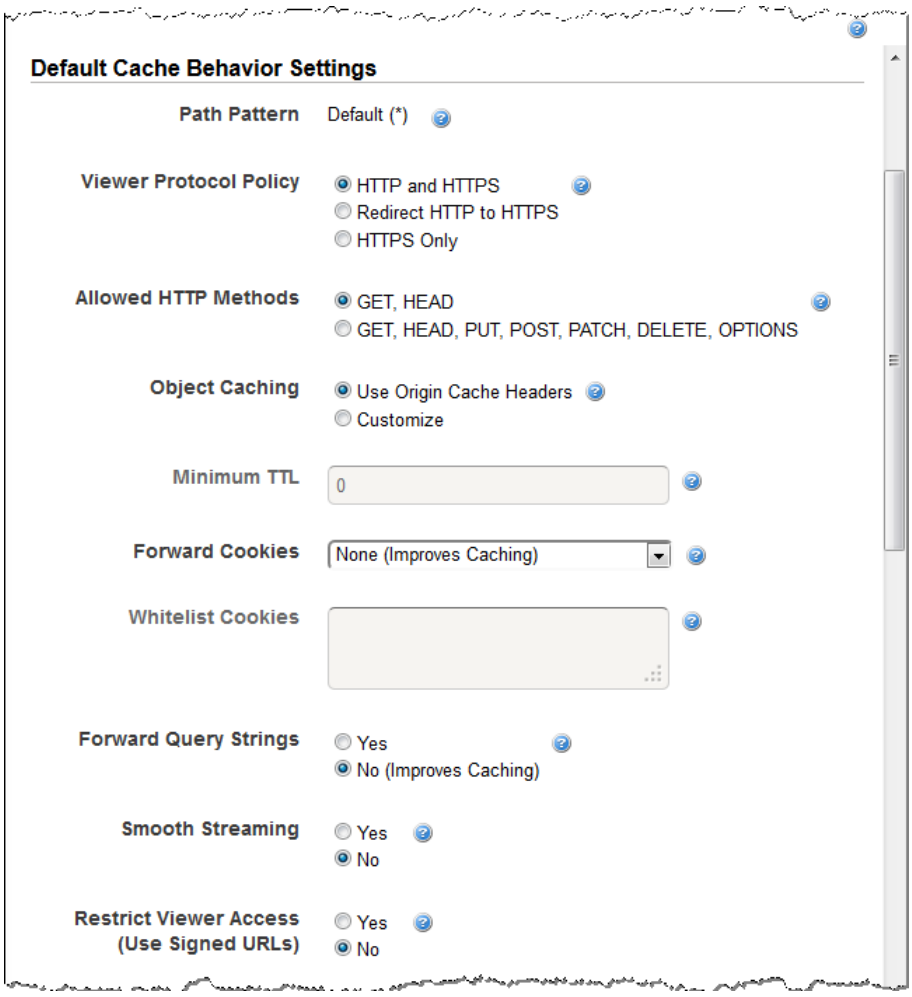
4. 选择您之前创建的 Amazon S3 存储桶。对于 Origin ID (源 ID) 和 Restrict Bucket Access (限制存储桶访问权限)，请接受默认值。



5. 在 Default Cache Behavior Settings (默认缓存行为设置) 下，接受默认值，CloudFront 将：
   - 将使用您的分配 CloudFront URL（例如，`http://d111111abcdef8.cloudfront.net/image.jpg`）的所有请求都转发到您在第 4 步中指定的 Amazon S3 存储桶。
   - 允许最终用户使用 HTTP 或 HTTPS 访问您的对象。
   - 响应对象请求。
   - 在 CloudFront 节点将您的对象缓存 24 小时。
   - 将对象请求转发到源时，排除 Cookie 和查询字符串参数（如有）。（Amazon S3 不处理 Cookie，只处理一组有限的查询字符串参数。）

- 允许每个人查看您的内容。

有关缓存行为选项的更多信息，请参阅缓存行为设置 (p. 33)。



6. 在 Distribution Details (分配详细信息) 下，输入适用的值：

   - Price Class: (价格级别:)选择与您想为 CloudFront 服务支付的最高价对应的价格级别。默认情况下，CloudFront 从所有 CloudFront 区域的节点提供您的对象。

     有关价格级别以及您选择的价格级别如何影响分配的 CloudFront 性能的更多信息，请转到选择 CloudFront 分配的价格级别 (p. 54)。有关 CloudFront 定价的信息，包括价格级别与 CloudFront 区域的对应关系，请转到 Amazon CloudFront 定价。
   - Alternate Domain Names (CNAMEs): (备用域名 (别名记录):) 可选。指定您想用于对象 URL 的一个或多个域名，来代替您创建分配时 CloudFront 指派的域名。例如，如果您希望对象的 URL：

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

     像这样：

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

     而不是这样：

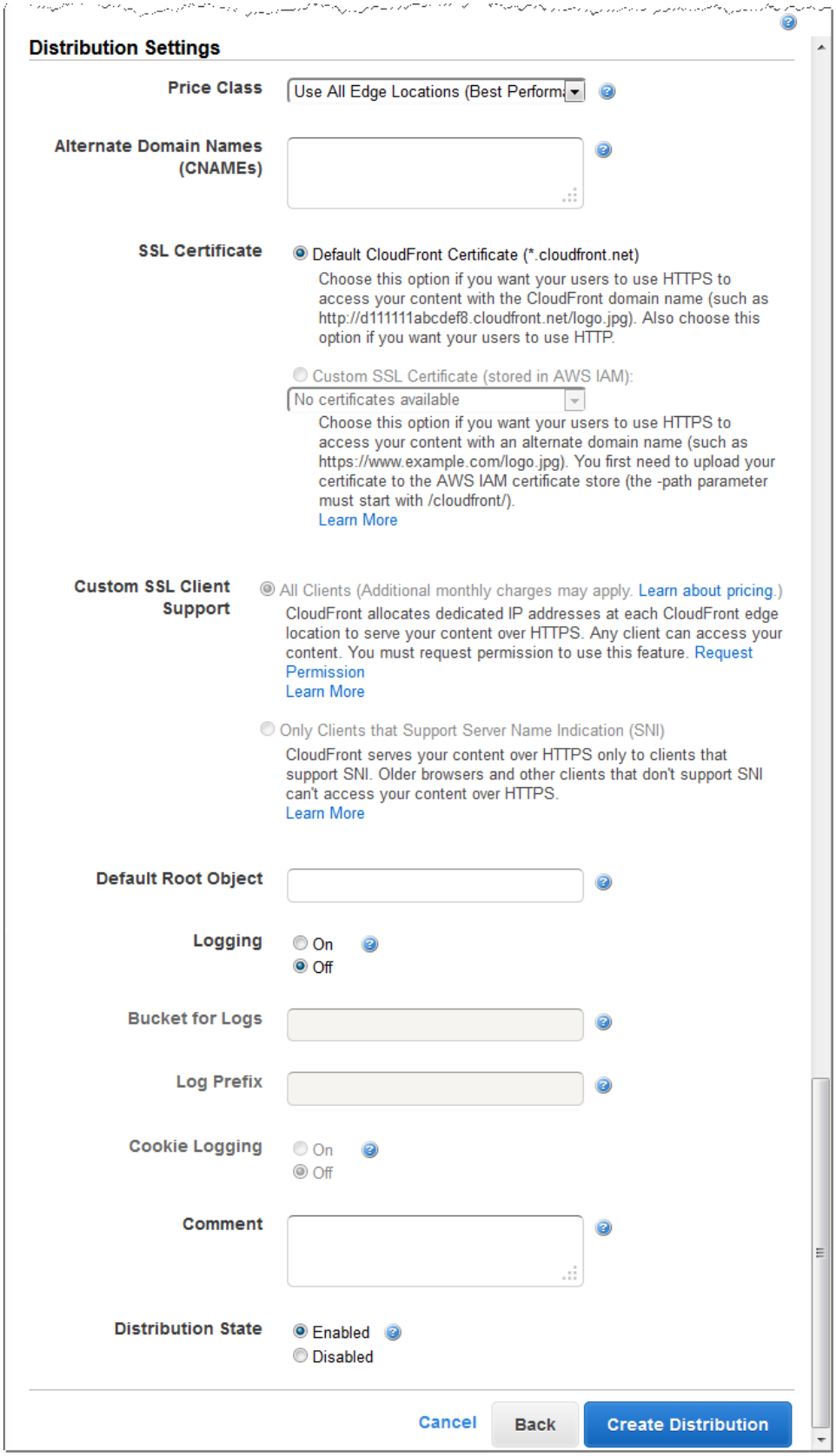
```
http://d111111abcdef8.cloudfront.net/images/image.jpg
```

您可为 `www.example.com` 创建别名记录 (CNAME)。您最多可为每个分配创建 10 个别名记录 (CNAME)。

Important

如果您将 `www.example.com` 的别名记录 (CNAME) 添加到您的分配，您还需要使用 DNS 服务创建（或更新）别名记录，以便将 `www.example.com` 查询发送到 `s5c39gqb8ow64r.cloudfront.net`。您必须具有权限，才能使用 DNS 服务提供商为域创建别名记录 (CNAME)。通常，这意味着您拥有该域，但您还可以为域的拥有者开发应用程序。有关别名记录 (CNAME) 的更多信息，请参阅 使用备用域名（别名记录）(p. 52)。

- Default Root Object: (默认根对象:) 可选。当查看者请求分配的根 URL (`http://www.example.com/`) 而不是分配中的对象 (`http://www.example.com/product-description.html`) 时，您希望 CloudFront 从您的源（例如，`index.html`）中请求的对象。指定一个默认根对象，以避免公开分配的内容。
- Logging: (日志记录:) 可选。如果您希望 CloudFront 记录有关每个对象请求的信息并将日志文件存储在 Amazon S3 存储桶中，请选择 On (打开)，然后指定存储桶并为日志文件名称指定可选的前缀。启用日志记录无需额外付费，但是，存储和访问文件将会产生常规 Amazon S3 费用。CloudFront 不自动删除日志，但您可以随时删除它们。
- Cookie Logging: (Cookie 日志记录:)在该示例中，我们使用 Amazon S3 作为对象的源，且 Amazon S3 不处理 Cookie，因此，我们建议您选择 Off (关闭) 作为 Cookie Logging (Cookie 日志记录) 的值。
- Comment: (评论:) 可选。输入您想与分配一起保存的任何评论。
- Distribution State: (分配状态:)如果您希望 CloudFront 在分配创建后立即开始处理请求，请选择 Enabled (启用)，或者，如果您不希望 CloudFront 在分配创建后开始处理请求，请选择 Disabled (禁用)。


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
Cancel
Back
Create Distribution

7. 单击 Create Distribution (创建分配)。
8. 在 CloudFront 创建了分配后，您的分配的 Status (状态) 列的值将从 InProgress (进行中) 更改为 Deployed (已部署)。如果您选择启用分配，随后它将可以处理请求。这应该需要不到 15 分钟的时间。

   CloudFront 指派给您的分配的域名将出现在分配列表中。（它同时也出现在选定分配的 General (常规) 选项卡上。）

第 4 步：测试链接

创建分配后，CloudFront 将会知道您的 Amazon S3 原始服务器的位置，而且您也知道与该分配相关联的域名。您可创建具有此域名的 Amazon S3 存储桶内容的链接，并让 CloudFront 提供该内容。

Note

在测试您的链接之前，您必须等待，直到分配的状态变为 Deployed (已部署)。

链接到您的对象

1. 将以下 HTML 复制到新文件中：

   - 用 CloudFront 指派给您的分配的域名替换 <domain name>。
   - 用 Amazon S3 存储桶中的文件名替换 <object name>。

```
<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>
```

   例如，如果您的域名为 `d111111abcdef8.cloudfront.net`，且对象名称为 `image.jpg`，则链接的 URL 将为：

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

   如果您的对象位于存储桶内的一个文件夹中，则将该文件夹包含在 URL 中。例如，如果 image.jpg 位于图像文件夹中，则 URL 为：

   http://d111111abcdef8.cloudfront.net/images/image.jpg
2. 将文本保存在具有 .html 文件扩展名的文件中。
3. 在浏览器中打开您的网页，以确保您可以看到您的内容。如果您无法看到内容，请确认您已正确执行了所有步骤。您还可以查看故障排除 (p. 154)中的提示。

浏览器返回带嵌入图像文件的页面，该文件由 CloudFront 确定适合提供对象的节点提供。

有关使用 CloudFront 的更多信息，请转到我从这里可以继续进行哪些内容？ (p. 249)。

创建 Web 分配

本节介绍如何创建 CloudFront Web 分配。有关 Web 分配的更多信息，请参阅使用 Web 分配 (p. 28)。

创建 Web 分配的任务列表

以下任务列表总结了创建 Web 分配的过程。

创建 Web 分配

1. 创建一个或多个 Amazon S3 存储桶或将 HTTP 服务器配置为您的原始服务器。源是您存储网页内容原始版本的位置。当 CloudFront 收到针对您的文件的请求时，它将转到源，以获取它在节点位置分配的文件。您可以将最多 10 个 Amazon S3 存储桶和 HTTP 服务器的任意组合用作您的原始服务器。

   如果您正在使用 Amazon S3，请注意，存储桶的名称必须全部小写，并且不能包含空格。

   如果您正在使用 Amazon EC2 服务器或其他自定义源，请查看使用 Amazon EC2 和其他自定义源的要求和建议 (p. 41)。
2. 将内容上传至您的原始服务器。如果您不希望限制使用 CloudFront 签署的 URL 对内容的访问，则可使对象公开可读。

   Caution

   您负责确保您的原始服务器的安全。您必须确保 CloudFront 有权限访问服务器，并确保安全设置是适当的，足以保护您的内容。

3. 创建 CloudFront Web 分配：
   - 有关使用 CloudFront 控制台创建 Web 分配的更多信息，请参阅使用 CloudFront 控制台创建 Web 分配 (p. 20)。
   - 有关使用 CloudFront API 创建 Web 分配的信息，请转到 *Amazon CloudFront API 参考* 中的 POST 分配。
4. 可选：如果您已使用 CloudFront 控制台创建了分配，则可为您的分配创建更多的缓存行为或源。有关更多信息，请参阅 使用 CloudFront 控制台列出、查看及更新 CloudFront 分配 (p. 55)。
5. 测试您的 Web 分配。有关更多信息，请参阅 测试您的 Web 分配 (p. 20)。

6. 开发您的网站或应用程序以使用在步骤 3 中创建分配后 CloudFront 返回的域名访问您的内容。例如，如果 CloudFront 返回 d111111abcdef8.cloudfront.net 作为您的分配的域名，则 Amazon S3 存储桶中或 HTTP 服务器上的根目录中的 `image.jpg` 文件的 URL 将是 `http://d111111abcdef8.cloudfront.net/image.jpg`。

   如果您在创建分配时指定了一个或多个备用域名（别名记录），则可使用您自己的域名。在这种情况下，`image.jpg` 的 URL 可能是 `http://www.example.com/image.jpg`。

   请注意以下几点：

   - 如果您想使用签署的 URL 来限制对内容的访问，请参阅通过 CloudFront 提供私有内容。 (p. 91)。
   - 如果您想提供压缩的内容，请参阅提供压缩文件 (p. 76)。
   - 有关 Amazon S3 的 CloudFront 请求和响应行为以及自定义源的信息，请参阅请求和响应行为 (p. 80)。

使用 CloudFront 控制台创建 Web 分配

以下步骤介绍了如何使用 CloudFront 控制台创建 Web 分配。如果您想使用 CloudFront API 创建 Web 分配，请转到 *Amazon CloudFront API 参考* 中的 POST 分配。

默认情况下，您总共可为每个 AWS 账户创建 100 个 Web 和 RTMP 分配。要申请更高限额，请转到 https://aws.amazon.com/support/createCase?type=service_limit_increase&serviceLimitIncreaseType=cloudfront-distributions。

使用 CloudFront 控制台创建 CloudFront Web 分配

1. 登录 AWS 管理控制台，并通过以下网址打开 Amazon CloudFront 控制台：https://console.aws.amazon.com/cloudfront/。
2. 单击 Create Distribution (创建分配)。
3. 在 Create Distribution Wizard (创建分配向导) 的第一页上，接受默认选择 Web，然后单击 Continue (继续)。
4. 指定分配的设置。有关更多信息，请参阅 您创建或更新 Web 分配时指定的值 (p. 29)。
5. 单击 Create Distribution (创建分配)。
6. 在 CloudFront 创建分配后，分配的 Status (状态) 列的值将从 InProgress (进行中) 改为 Deployed (已部署)。如果您选择启用分配，随后它将可以处理请求。这应该需要不到 15 分钟的时间。

   CloudFront 指派给您的分配的域名将出现在分配列表中。（它同时也出现在选定分配的 General (常规) 选项卡上。）

在部署您的分配时，请确认您可使用新的 CloudFront URL 或别名记录访问您的内容。有关更多信息，请参阅 测试您的 Web 分配 (p. 20)。

测试您的 Web 分配

创建分配后，CloudFront 将会知道您的原始服务器的位置，而且您也知道与该分配相关联的域名。您可使用 CloudFront 域名为您的对象创建链接，CloudFront 会将对象提供给您的网页或应用程序。

Note

在可以测试您的链接之前，您必须等待，直到分配的状态变为 Deployed (已部署)。

为 Web 分配中的对象创建链接

1. 将以下 HTML 代码复制到新文件中，用您的分配域名替换 *domain-name*，然后用您的对象名称替换 *object-name*。

```
<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>
```

   例如，如果您的域名为 d111111abcdef8.cloudfront.net，且对象名称为 image.jpg，则链接的 URL 将为：

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

   如果您的对象是在原始服务器上的一个文件夹中，则此文件夹也必须包含在 URL 中。例如，如果 image.jpg 位于原始服务器上的图像文件夹中，则 URL 为：

   http://d111111abcdef8.cloudfront.net/images/image.jpg
2. 将 HTML 代码保存在具有 .html 文件扩展名的文件中。
3. 在浏览器中打开您的网页，以确保您可以看到您的对象。

浏览器返回带嵌入图像文件的页面，该文件由 CloudFront 确定适合提供对象的节点提供。

创建 RTMP 分配

本节说明如何配置按需流媒体传输。如果您使用 Adobe Flash Player、Flowplayer 或 JW Player 作为您的媒体播放器，请参阅适用的教程：

- 使用 CloudFront 和 Adobe Flash Player 的按需视频流 (p. 234)
- 使用 CloudFront 和 Flowplayer for Adobe Flash 的按需视频流 (p. 239)
- 使用 CloudFront 和 JW Player 的按需视频流 (p. 244)

流式处理媒体文件的任务列表

以下任务列表总结了创建 Web 分配的过程。

创建 RTMP 分配

1. 为您的媒体文件创建 Amazon S3 存储桶。如果您为媒体播放器使用不同的 Amazon S3 存储桶，则也要为媒体播放器文件创建 Amazon S3 存储桶。

   存储桶的名称必须全部小写，并且不能包含空格。
2. 选择并配置媒体播放器，以播放您的媒体文件。有关更多信息，请参阅媒体播放器的文档。
3. 将您的媒体播放器文件上传到您希望 CloudFront 从中获取文件的源。如果您使用 Amazon S3 存储桶作为媒体播放器的源，则可使文件（非存储桶）公开可读。
4. 为您的媒体播放器创建 Web 分配。（您也可使用现有的分配。）有关更多信息，请参阅 *创建 Web 分配* (p. 19)。
5. 将您的媒体文件上传到您为媒体文件创建的 Amazon S3 存储桶，并使内容（非存储桶）公开可读。

   Important

   Flash Video 容器中的媒体文件必须包括 .flv 文件扩展名，否则，媒体将不会流式处理。

   您可将媒体播放器文件和媒体文件放在同一存储桶中。
6. 为您的媒体文件创建 RTMP 分配：

   - 有关使用 CloudFront 控制台创建 Web 分配的更多信息，请参阅使用 CloudFront 控制台创建 RTMP 分配 (p. 23)。
   - 有关使用 CloudFront API 创建 Web 分配的信息，请转到 *Amazon CloudFront API 参考* 中的 POST 流分配。

7. 配置您的媒体播放器。有关更多信息，请参阅 配置媒体播放器 (p. 50)。

如果播放内容时出现问题，请参阅对 RTMP 分配进行故障排除 (p. 51)。

使用 CloudFront 控制台创建 RTMP 分配

以下步骤介绍了如何使用 CloudFront 控制台创建 RTMP 分配。如果您想使用 CloudFront API 创建 RTMP 分配，请转到 *Amazon CloudFront API 参考* 中的 POST 流分配。

默认情况下，您总共可为每个 AWS 账户创建 100 个 Web 和 RTMP 分配。要申请更高限额，请转到 https://aws.amazon.com/support/createCase?type=service_limit_increase&serviceLimitIncreaseType=cloudfront-distributions。

使用 CloudFront 控制台创建 RTMP 分配

1. 登录 AWS 管理控制台，并通过以下网址打开 Amazon CloudFront 控制台：https://console.aws.amazon.com/cloudfront/。
2. 单击 Create Distribution (创建分配)。
3. 在 Create Distribution Wizard (创建分配向导) 的第一页上，单击 RTMP，然后单击 Continue (继续)。
4. 指定分配的设置。有关更多信息，请参阅 您创建或更新 RTMP 分配时指定的值 (p. 46)。
5. 单击 Create Distribution (创建分配)。
6. 在 CloudFront 创建分配后，分配的 Status (状态) 列的值将从 InProgress (进行中) 改为 Deployed (已部署)。如果您选择启用分配，随后它将可以处理请求。这应该需要不到 15 分钟的时间。

   CloudFront 指派给您的分配的域名将出现在分配列表中。域名同时也出现在选定分配的 General (常规) 选项卡上。

将 CloudFront 与 Amazon S3 结合使用

如果目前您使用别名记录 (CNAME) 从您的 Amazon S3 存储桶分发内容，则您可以按照以下过程迁移到 CloudFront，同时不会产生中断。

如果您的对象存储在 Amazon S3 中，那么您可以让用户直接从 Amazon S3 存储桶中获取这些对象，也可将 CloudFront 配置为从 Amazon S3 获取这些对象，并让用户从 CloudFront 获取它们。

如果您的用户频繁访问您的对象，您可以通过添加 CloudFront 来降低传输这些对象所耗费的成本，因为在使用量较大的情况下，CloudFront 数据传输价格低于 Amazon S3 数据传输价格。此外，使用 CloudFront 时下载速度也比仅使用 Amazon S3 时更快，因为您的对象存储在离您用户更近的位置。

Important

CloudFront 目前不支持 Amazon S3 跨源资源共享 (CORS)。

如果您目前使用您自己的域名（例如 example.com）而不是使用 Amazon S3 存储桶的域名（例如 MyAWSBucket.s3.amazonaws.com）直接从您的 Amazon S3 存储桶分发内容，那么您可以按照以下过程添加 CloudFront，同时也不会造成中断。

当您已经在从 Amazon S3 分发内容时添加 CloudFront 的过程

1. 使用相应主题中介绍的过程创建一项 CloudFront 分配：

   - 创建 Web 分配 (p. 19)
   - 创建 RTMP 分配 (p. 22)

   当您创建分配时，请指定您的 Amazon S3 存储桶的名称作为原始服务器。

   Important

   要使您的存储桶能够与 CloudFront 一起使用，其名称必须符合 DNS 命名要求。有关详细信息，请参阅 *Amazon Simple Storage Service 开发者指南* 中的 Bucket Restrictions and Limitations。

   如果您对 Amazon S3 使用的是别名记录，请也为您的分配指定别名记录。

2. 创建一个测试网页并在该网页中包含您的 Amazon S3 存储桶中公开可读对象的链接，然后测试这些链接。对于此初步测试，请在对象 URL 中使用此项分配的 CloudFront 域名，例如 `http://d111111abcdef8.cloudfront.net/images/image.jpg`。

   有关 CloudFront URL 格式的更多信息，请参阅 CloudFront 对象的 URL 格式 (p. 57)。

3. 如果您使用的是 Amazon S3 别名记录，您的应用程序将使用您的域名（例如，example.com）来引用您 Amazon S3 存储桶中的对象，而不是使用您的存储桶的名称（例如，myawsbucket.s3.amazonaws.com）。要继续使用您的域名来引用对象，而不是使用您的分配的 CloudFront 域名（例如，d111111abcdef8.cloudfront.net），您需要与您的 DNS 服务提供商联系来更新您的设置。

   为使您的 Amazon S3 别名记录正常工作，您的 DNS 服务提供商有一个与您的域对应的别名记录资源记录集，该资源记录集目前将最终用户对该域的查询传送到您的 Amazon S3 存储桶。当有人请求下面的对象时：

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

   该请求的传送路线将自动调整，请求者所看到的对象将为：

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

   要将查询传送到您的 CloudFront 分配而不是您的 Amazon S3 存储桶，您需要按照 DNS 服务提供商提供的方法来更新与您的域对应的别名记录资源记录集。这条更新后的别名记录将开始将 DNS 查询从您的域重定向到您的分配的 CloudFront 域名。有关更多信息，请参阅 DNS 服务提供商提供的文档。

   Note

   如果您使用 Route 53 作为您的 DNS 服务，那么要了解有关如何更新别名记录资源记录集的信息，请访问创建、更改和删除资源记录集。

   有关对 CloudFront 使用别名记录的更多信息，请参阅使用备用域名（别名记录）(p. 52)。

   在您更新别名记录资源记录集后，最多可能需要 72 个小时的时间才能将所做更改传播到整个 DNS 系统，不过通常传播速度要比这快。在此期间，针对您内容的一些请求将继续传送到您的 Amazon S3 存储桶，其他请求将传送到 CloudFront。

使用分配

Topics

CloudFront 分配确定您希望 CloudFront 通过其节点分配的文件并控制您希望 CloudFront 如何分配文件的各种细节。当您创建分配时，可指定配置设置，如：

- 您希望 CloudFront 在哪里获得其分配给节点的文件 — Amazon S3 存储桶还是 Web 服务器。
- 您是否希望文件可提供给每一个人还是您想限制对选定用户的访问。
- 您是否希望 CloudFront 创建访问日志。

对 CloudFront API 的更改

从 CloudFront API 2012-05-05 版本开始，我们对您创建或更新 Web 分配或 RTMP 分配时以及您使对象失效时，包含在请求主体中的 XML 文档的格式做了重大更改。对于 API 的以前版本，我们发现很容易意外删除接受多个值的元素的一个或多个值，例如，别名记录 (CNAME) 和可信签署人。我们对 2012-05-05 版本所做的更改是为了预防此类意外删除情况，并在元素的值数量与您实际指定的值数量不匹配时通知您。例如，您在 `Quantity` 元素中指定的值数量与您实际指定的值数量不匹配。

请注意以下关于使用 2012-05-05 API 版本或更高版本处理使用早期 API 版本创建的 Web 和 RTMP 分配的内容：

- 您不能使用早于 2012-05-05 的 API 版本更新使用 2012-05-05 或更高 CloudFront API 版本创建或更新的 Web 分配。
- 您可使用新的 API 版本获取分配列表、有关分配的信息或分配配置。CloudFront 将以新 XML 格式返回 XML 文档。

- 要更新使用早期 API 版本创建的分配，请使用 GET 分配或 GET 流分配的 2012-05-05 或更高版本获取新 XML 格式的 XML 文档，根据情况更改数据，然后使用 PUT 分配配置或 PUT 流分配配置的 2012-05-05 或更高版本将更改提交给 CloudFront。
- 您可使用新的 API 删除使用早期 API 版本创建的分配。分配必须已经禁用。

Web 和 RTMP 分配概述

Note

默认情况下，您总共可为每个 AWS 账户创建 100 个 Web 和 RTMP 分配。要申请更高限额，请转到 https://aws.amazon.com/support/createCase?type=service_limit_increase&serviceLimitIncreaseType=cloudfront-distributions。

您可以为每个分配提供无限个文件。

Web 分配使用 HTTP 或 HTTPS 提供静态和动态内容，例如 .html、.css、.php 和图像文件。Web 分配从您指定的位置（称为源）获取文件。每个源是 Amazon S3 存储桶或 HTTP 服务器，例如 Web 服务器。您最多可以指定任意组合的 10 个 Amazon S3 存储桶和/或 HTTP 服务器作为您的源。

您还可以使用 Web 分配来分配一个实时事件，例如实时会议或音乐会。对于实时流，您使用 AWS CloudFormation 堆栈自动创建分配。有关更多信息，请参阅 CloudFront 教程 (p. 165) 中适用的实时流教程。

有关 Web 分配工作机制的更多信息，包括您在创建 Web 分配时指定的值，请参阅 使用 Web 分配 (p. 28)。有关创建 Web 分配的信息，请参阅 创建 Web 分配 (p. 19)。

RTMP 分配使用 Adobe Media Server 和 Adobe 实时消息协议 (RTMP) 流式处理媒体文件。RTMP 分配必须使用 Amazon S3 存储桶作为源。

有关您在创建 RTMP 分配时指定的值的信息，请参阅 使用 RTMP 分配 (p. 43)。有关创建 RTMP 分配的信息，请参阅 创建 RTMP 分配 (p. 22)。

对分配的操作

下表列出了您可对分配执行的操作，并提供了有关如何使用 CloudFront 控制台和 CloudFront API 执行操作的相应文档的链接。

操作	使用 CloudFront 控制台	使用 CloudFront API：Web 分配	使用 CloudFront API：RTMP 分配
创建分配	Web 分配：请参阅 创建 Web 分配 (p. 19) RTMP 分配：请参阅 创建 RTMP 分配 (p. 22)	请转到 POST 分配	请转到 POST 流分配
列出您的分配	请参阅 列出、查看及更新 CloudFront 分配 (p. 55)	请转到 GET 分配列表	请转到 GET 流分配列表
获取有关分配的所有信息	请参阅 列出、查看及更新 CloudFront 分配 (p. 55)	请转到 GET 分配	请转到 GET 流分配

操作	使用 CloudFront 控制台	使用 CloudFront API：Web 分配	使用 CloudFront API：RTMP 分配
获取分配配置	请参阅 列出、查看及更新 CloudFront 分配 (p. 55)	请转到 GET 分配配置	请转到 GET 流分配配置
更新分配	请参阅 列出、查看及更新 CloudFront 分配 (p. 55)	请转到 PUT 分配配置	请转到 PUT 流分配配置
删除分配	请参阅 删除分配 (p. 56)	请转到 DELETE 分配	请转到 DELETE 流分配

使用 Web 分配

Topics

本节说明如何配置和管理 CloudFront Web 分配。有关分配的基本介绍，请参阅 使用分配 (p. 26)。有关 CloudFront RTMP 分配的信息，请参阅 使用 RTMP 分配 (p. 43)。

为 Web 分配使用 Amazon S3 源和自定义源

当您创建 Web 分配时，指定 CloudFront 将对分配至节点的文件的请求发送至何处。CloudFront 支持使用 Amazon S3 存储桶和 HTTP 服务器（例如 Web 服务器）作为源。

为您的源使用 Amazon S3 存储桶

当您使用 Amazon S3 作为分配的源时，请将您想要 CloudFront 提供的任何对象放在 Amazon S3 存储桶中。您可以使用 Amazon S3 支持的任何的方法将对象置入 Amazon S3，例如 Amazon S3 控制台或 API，或者第三方工具。您可以在用于存储对象的存储桶中创建一个层次结构，就如您使用任何其他 Amazon S3 存储桶一样。

使用现有的 Amazon S3 存储桶作为您的 CloudFront 原始服务器不会以任何方式更改存储桶；您仍然可以按标准 Amazon S3 价格照常存储和访问 Amazon S3 对象。在存储桶中存储对象会产生常规的 Amazon S3 费用。有关 CloudFront 使用费的更多信息，请参阅 CloudFront 计费和使用 (p. 6)。

Important

要使您的存储桶能够与 CloudFront 一起使用，其名称必须符合 DNS 命名要求。有关详细信息，请参阅 *Amazon Simple Storage Service 开发者指南* 中的 Bucket Restrictions and Limitations。

当您指定希望 CloudFront 从中获取对象的 Amazon S3 存储桶时，如何指定存储桶的名称取决于您是否将存储桶配置为网站终端节点：

存储桶没有配置为网站终端节点
使用以下格式：

bucket-name.s3.amazonaws.com

当您以该格式指定存储桶名称时，可以使用以下 CloudFront 功能：

- 将 CloudFront 配置为使用 SSL 与您的 Amazon S3 存储桶通信。有关更多信息，请参阅 使用 HTTPS 连接访问您的对象 (p. 134)。
- 使用原始访问标识要求您的用户使用 CloudFront URL（而非使用 Amazon S3 URL）访问您的内容。有关更多信息，请参阅 使用原始访问标识限制访问您的 Amazon S3 内容 (p. 95)。
- 通过向 CloudFront 提交 POST 和 PUT 请求，更新您的存储桶内容。有关更多信息，请参阅 CloudFront 如何处理请求及如何将请求转发给您的 Amazon S3 原始服务器 (p. 80) 主题中的 HTTP 方法 (p. 81)。

存储桶配置为网站终端节点
为您的存储桶输入 Amazon S3 静态网站托管终端节点。该值显示在 Amazon S3 控制台 Properties (属性) 页面的 Static Website Hosting (静态网站托管) 下。

当您以该格式指定存储桶名称时，可使用 Amazon S3 重定向和 Amazon S3 自定义错误文档。（CloudFront 也提供自定义错误页面。有关更多信息，请参阅 自定义错误响应 (p. 70)。）有关 Amazon S3 功能的更多信息，请参阅 Amazon S3 文档。

请勿使用以下格式指定存储桶：

- Amazon S3 路径样式，s3.amazonaws.com/*bucket-name*
- Amazon S3 别名记录（如果有）

使用 Amazon EC2 或其他自定义源

自定义源是 HTTP 服务器，例如，Web 服务器。HTTP 服务器可以是 Amazon EC2 实例或您私下管理的 HTTP 服务器。当您使用自定义源时，可指定服务器的 DNS 名称、HTTP 与 HTTPS 端口及您希望 CloudFront 从您的源中获取对象时使用的协议。

当您使用自定义源时，除以下功能外，大部分 CloudFront 功能受到支持：

- RTMP 分配— 不支持。
- 私有内容— 虽然您可以使用签名 URL 从自定义源分配内容，但要使 CloudFront 访问自定义源，源必须可公共访问。有关更多信息，请参阅 通过 CloudFront 提供私有内容。 (p. 91)。

有关使用自定义源时的要求和建议的信息，请参阅 使用 Amazon EC2 和其他自定义源的要求和建议 (p. 41)。

您创建或更新 Web 分配时指定的值

当您创建新的 Web 分配或更新现有的分配时，请指定以下值。有关使用 CloudFront 控制台创建或更新 Web 分配的信息，请参阅相关主题：

- 创建 Web 分配 (p. 19)
- 列出、查看及更新 CloudFront 分配 (p. 55)

传输方式 (p. 31)

源设置 (p. 31)

- 源域名 (p. 31)
- 源 ID (p. 32)

- 国家/地区 (p. 40)

传输方式

当您创建分配时，请指定传输方式。对于 Web 分配，该值始终为 Web。您不能更改现有分配的传输方式。

源设置

当您创建或更新分配时，请提供您用来存储 Web 内容的原始版本的一个或多个位置（又称为源）。CloudFront 从您的源获取您的 Web 内容并通过全球的边缘服务器网络提供给查看者。每个源是 Amazon S3 存储桶或 HTTP 服务器，例如 Web 服务器。

默认情况下，您最多可为一个分配创建 10 个源。要申请更高限额，请转到 https://aws.amazon.com/support/createCase?type=service_limit_increase&serviceLimitIncreaseType=cloudfront-distributions。

如果您想删除源，首先必须编辑或删除与该源关联的缓存行为。

Caution

如果您删除源，请确认该源先前提供的文件在另一源中可用，且您的缓存行为正将对这些文件的请求路由到新的源。

当您创建或更新分配时，请为每个源指定以下值。

源域名

您希望 CloudFront 从中获取该源的对象的 Amazon S3 存储桶或 HTTP 服务器的 DNS 域名，例如 `myawsbucket.s3.amazonaws.com` 或 `www.example.com`。

如果您的源是 HTTP 服务器，请键入该资源的域名。文件必须公开可读。

如果您的源是 Amazon S3 存储桶，请在 CloudFront 控制台中，单击 Origin Domain Name (源域名) 字段，将显示一个列表，枚举与当前 AWS 账户关联的 Amazon S3 存储桶。请注意以下几点：

- 如果存储桶配置为网站，请为您的存储桶输入 Amazon S3 静态网站托管终端节点；请勿从 Origin Domain Name (源域名) 字段的列表中选择存储桶名称。该静态网站托管终端节点显示在 Amazon S3 控制台的 Properties (属性) 页面中的 Static Website Hosting (静态网站托管) 下。
- 如果您要从不同的 AWS 账户使用存储桶，而且该存储桶没有配置为网站，请按以下格式键入名称：

  `bucket-name``.s3.amazonaws.com`
- 文件必须公开可读，除非您使用 CloudFront 原始访问标识确保 Amazon S3 中内容的安全。有关更多信息，请参阅 使用原始访问标识限制访问您的 Amazon S3 内容 (p. 95)。

Important

如果源是 Amazon S3 存储桶，则存储桶名称必须符合 DNS 命名要求。有关详细信息，请参阅 *Amazon Simple Storage Service 开发者指南* 中的 Bucket Restrictions and Limitations。

当您更改源的 Origin Domain Name (源域名) 值时，CloudFront 会立即将该更改复制到 CloudFront 节点。在给定节点中更新分配配置前，CloudFront 会继续向以前的 HTTP 服务器或 Amazon S3 存储桶转发请求。该节点中的分配配置一经更新，CloudFront 就开始向新的 HTTP 服务器或 Amazon S3 存储桶转发请求。

更改源并不需要 CloudFront 用新源中的对象重新填充边缘缓存。只要您应用程序中的查看者请求没有更改，CloudFront 将继续提供边缘缓存中已有的对象，直至每个对象的 TTL 过期或很少被请求的对象被淘汰。

源 ID

该分配中用以唯一标识源的字符串。如果您创建默认缓存行为之外的缓存行为，请使用在此指定的源 ID 标识您希望 CloudFront 在请求与该缓存行为的路径模式相匹配时将请求路由至的源。有关更多信息，请参阅 缓存行为设置 (p. 33)。

限制存储桶访问（仅限 Amazon S3）

如果您希望要求最终用户仅使用 CloudFront URL（而非使用 Amazon S3 URL）访问 Amazon S3 存储桶中的对象，请单击 Yes (是)。然后指定适用的值。

如果您希望最终用户能够使用 CloudFront URL 或 Amazon S3 URL 访问对象，请单击 No (否)。

有关更多信息，请参阅 使用原始访问标识限制访问您的 Amazon S3 内容 (p. 95)。

原始访问标识（仅限 Amazon S3）

如果您对 Restrict Bucket Access (限制存储桶访问) 选择了 Yes (是)，请选择创建新原始访问标识还是使用与您的 AWS 账户关联的现有原始访问标识。如果您已经有原始访问标识，我们建议您重用该标识以便简化维护。有关原始访问标识的更多信息，请参阅 使用原始访问标识限制访问您的 Amazon S3 内容 (p. 95)。

新标识的注释（仅限 Amazon S3）

如果您对 Origin Access Identity (原始访问标识) 选择了 Create a New Identity (创建新标识)，请输入标识新原始访问标识的注释。当您创建该分配时，CloudFront 将创建该原始访问标识。

您的标识（仅限 Amazon S3）

如果您对 Origin Access Identity (原始访问标识) 选择了 Use an Existing Identity (使用现有标识)，请选择您想要使用的原始访问标识。您不能使用与另一 AWS 账户关联的原始访问标识。

授予对存储桶的读取权限（仅限 Amazon S3）

如果您希望 CloudFront 自动向原始访问标识授予读取您的 Amazon S3 存储桶中的对象的权限，请单击 Yes, Update Bucket Policy (是，更新存储桶策略)。

Important

如果您单击 Yes, Update Bucket Policy (是，更新存储桶策略)，CloudFront 会更新存储桶策略，以向指定的原始访问标识授予读取您的存储桶中对象的权限。然而，CloudFront 并不删除存储桶策略中的现有权限或对个别对象的权限。如果用户目前具有使用 Amazon S3 URL 访问您的存储桶中对象的权限，在 CloudFront 更新您的存储桶策略后，他们仍然具有该权限。要查看或更改现有存储桶策略以及对存储桶中对象的现有权限，请使用 Amazon S3 提供的方法。有关更多信息，请参阅 向原始访问标识授予读取 Amazon S3 存储桶中对象的权限 (p. 97)。

如果您希望手动更新权限，例如您希望更新对象的 ACL 而非更新存储桶权限，请单击 No, I will Update Permissions (否，我将更新权限)。

原始协议策略（仅限 Amazon EC2 和其他自定义源）

您希望 CloudFront 从您的原始服务器获取对象时使用的协议策略。如果您指定 HTTP Only (仅 HTTP)，则 CloudFront 仅使用 HTTP 访问源。

如果您指定 Match Viewer (匹配查看者)，则 CloudFront 根据查看者请求的协议使用 HTTP 或 HTTPS 从您的源获取对象。请注意，CloudFront 仅缓存对象一次，即使查看器同时使用 HTTP 和 HTTPS 协议发出请求也是如此。

HTTP 端口（仅限 Amazon EC2 和其他自定义源）

可选。自定义源侦听的 HTTP 端口。有效值包括端口 80、443 和 1024 到 65535。默认值是端口 80。

HTTPS 端口（仅限 Amazon EC2 和其他自定义源）

可选。自定义源侦听的 HTTPS 端口。有效值包括端口 80、443 和 1024 到 65535。默认值是端口 443。

缓存行为设置

通过缓存行为可为您网站上文件的特定 URL 路径模式配置各种 CloudFront 功能。例如，一个缓存行为可能适用于您用作 CloudFront 原始服务器的 Web 服务器上 `images` 目录中所有的 `.jpg` 文件。您可为每个缓存行为配置的功能包括：

- 路径模式。
- 如果您已经为您的 CloudFront 分配配置了多个源，您希望 CloudFront 将您的请求转发至哪个源。
- 是否将查询字符串转发到您的源。
- 访问指定文件是否需要签名 URL。
- 是否要求最终用户使用 HTTPS 访问这些文件。
- 这些文件保留在 CloudFront 缓存中的最小时长，不管您的源添加到文件中任何 `Cache-Control` 标头的值。

当您创建新分配时，请为默认缓存行为指定设置，这将自动把所有请求转发到您创建分配时指定的源。在您创建分配后，可创建其他缓存行为，定义 CloudFront 在收到与路径模式相匹配的对象请求时如何响应，例如 `*.jpg`。如果您创建其他缓存行为，默认缓存行为将始终是最后处理的。其他缓存行为按照在 CloudFront 控制台中的列出顺序进行处理；如果您使用 CloudFront API，则按照在分配的 `DistributionConfig` 元素中的列出顺序进行处理。有关更多信息，请参阅 路径模式 (p. 33)。

当您创建缓存行为时，请指定您希望 CloudFront 从中获取对象的一个源。因此，如果您希望 CloudFront 分配您所有源中的对象，则必须至少具有与您所拥有的源一样多的缓存行为（包括默认缓存行为）。例如，您有两个源，但只有一个默认缓存行为，该默认缓存行为将导致 CloudFront 只从一个源获取对象，而永远不会使用另一个源。

默认情况下，您最多可为一个分配创建 10 个缓存行为。要申请更高限额，请转到 https://aws.amazon.com/support/createCase?type=service_limit_increase&serviceLimitIncreaseType=cloudfront-distributions。

路径模式

Note

当您创建新分配时，Path Pattern (路径模式) 的值默认为 *，而且不能更改。该值会导致 CloudFront 将对您对象的所有请求转发到您在 源域名 (p. 31) 字段中指定的源。

路径模式（例如，`/images/*.jpg`）指定您希望适用该缓存行为的请求。CloudFront 接收到最终用户请求时，会按照缓存行为在分配中的列出顺序，将请求路径与路径模式进行比较。第一个匹配者决定了适用于该请求的缓存行为。例如，您有三个具有以下三种路径模式的缓存行为，顺序如下：

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

对文件 /images/sample.gif 的请求不符合第一个路径模式，因此，关联的缓存行为不适用于该请求。该文件符合第二个路径模式，因此，与第二个路径模式关联的缓存行为适用（即使该请求与第三个路径模式也匹配）。

Caution

谨慎定义路径模式及其顺序，否则可能会向最终用户提供非预期的内容访问权。例如，假设某一请求与两项缓存行为的路径模式匹配。第一项缓存行为不要求使用签名 URL，而第二项缓存行为则要求使用签名 URL。最终用户将无需使用签名 URL 即可访问对象，因为 CloudFront 会处理与第一个匹配项关联的缓存行为。

您指定的路径适用于对指定目录以及该指定目录下所有子目录中所有文件的请求。例如，/images 目录包含 product1 和 product2 子目录，路径模式 /images/*.jpg 将适用于对 /images、/images/product1 和 /images/product2 目录中任何 .jpg 文件的请求。如果您希望将不同的缓存行为应用于 /images/product1 目录中的文件，而非 /images 和 /images/product2 目录中的文件，请为 /images/product1 创建单独的缓存行为，并将该缓存行为移到 /images 目录的缓存行为上方（前面）的位置。

您可在路径模式中使用以下通配符：

- * 匹配 0 或更多字符。
- ? 精准匹配 1 个字符。

以下示例展示了通配符的作用：

路径模式	与路径模式匹配的文件
/*.jpg	所有 .jpg 文件
/images/*.jpg	/images 目录及 /images 目录下子目录中所有的 .jpg 文件
/a*.jpg	• 文件名以 a 开头的所有 .jpg 文件，例如，apple.jpg 和 appalachian_trail_2012_05_21.jpg • 文件路径以 a 开头的所有 .jpg 文件，例如，/abra/cadabra/magic.jpg。
/a??.jpg	文件名以 a 开头且随后紧跟两个其他字符的所有 .jpg 文件，例如，ant.jpg 和 abe.jpg
/*.doc*	文件扩展名以 .doc 开头的所有文件，例如，.doc、.docx 和 .docm 文件。在这种情况下，您不能使用路径模式 *.doc?，因为该路径模式将不适用于对 .doc 文件的请求；? 通配符精确替换一个字符。

默认缓存行为的路径模式是 *（所有文件），而且不能更改。如果对象请求与任何其他缓存行为的路径模式都不匹配，CloudFront 将应用您在默认缓存行为中指定的行为。

路径模式的长度上限是 255 个字符。该值可包含以下任何字符：

- A-Z，a-z
- 0-9
- _ - .* $ / ~ " ' @ : +
- &（传递和返回为 &）

源（仅限现有分配）

现有源的 Origin ID (源 ID) 值。该值标识当请求（例如 http://example.com/logo.jpg）与缓存行为（例如 *.jpg）或默认缓存行为 (*) 的路径模式匹配时，您希望 CloudFront 将请求路由至的源。

查看者协议策略

您希望查看者用于访问 Origin (源) 指定的源中内容的协议策略。如果您指定 HTTP and HTTPS (HTTP 和 HTTPS)，则查看者可使用两种协议。如果您指定 HTTPS Only (仅 HTTPS)，则查看者只能使用 HTTPS 访问您的内容。有关更多信息，请参阅 使用 HTTPS 连接访问您的对象 (p. 134)。

允许的 HTTP 方法

指定您希望 CloudFront 处理哪些方法并转发到您的源：

- GET、HEAD：您只能使用 CloudFront 获取您源中的对象或获取对象标头。
- GET、HEAD、PUT、POST、PATCH、DELETE、OPTIONS：您可以使用 CloudFront 获取、添加、更新和删除对象以及获取对象标头。此外，您可以执行其他 POST 操作，例如从 Web 表格提交数据。

Note

CloudFront 会缓存对 GET 和 HEAD 请求的响应。CloudFront 不会缓存对使用其他方法的请求的响应。

Caution

如果您选择 GET、HEAD、PUT、POST、PATCH、DELETE、OPTIONS，可能需要限制对您的 Amazon S3 存储桶或自定义源的访问，以使用户无法执行您不希望他们执行的操作：

- 如果您要使用 Amazon S3 作为您的分配的源：创建 CloudFront 原始访问标识以限制对您的 Amazon S3 内容的访问，并授予原始访问标识适用的权限。例如，您将 CloudFront 配置为接受并转发这些方法*只是*因为您想要使用 PUT，则仍必须配置 Amazon S3 存储桶策略或 ACL 以适当处理 DELETE 请求。有关更多信息，请参阅 使用原始访问标识限制访问您的 Amazon S3 内容 (p. 95)。
- 如果您使用自定义源：配置原始服务器以处理所有方法。例如，您将 CloudFront 配置为接受并转发这些方法*只是*因为您想要使用 POST，则仍必须配置您的原始服务器以适当处理 DELETE 请求。

对象缓存

如果您的原始服务器向您的对象添加 `Cache-Control` 标头以控制对象保留在 CloudFront 缓存中的时间，请选择 Use Origin Cache Headers (使用原始缓存标头)。

要指定您的对象保留在 CloudFront 缓存中的最短时间，而不管 `Cache-Control` 标头，请选择 Customize (自定义)。然后，在 Minimum TTL (最小 TTL) 字段中，指定您希望对象保留在 CloudFront 缓存中的最短时间（秒），即使 `Cache-Control` 标头中指定了较低的值。

最小 TTL

您希望在 CloudFront 查询您的源以了解对象是否已更新之前，对象保留在 CloudFront 缓存中的最小时长。有关更多信息，请参阅 指定对象在 CloudFront 边缘缓存中的保留时间（过期）(p. 63)。

转发 Cookie（仅限 Amazon EC2 和其他自定义源）

指定您是否希望 CloudFront 将 Cookie 转发到您的原始服务器；如果是，转发哪些 Cookie。如果您选择仅转发选定的 Cookie (Cookie 白名单)，请在 Whitelist Cookies (Cookie 白名单) 字段中输入 Cookie 的名称。如果您选择 All (全部)，CloudFront 将转发所有 Cookie，而不管您的应用程序使用多少。

Amazon S3 不处理 Cookie，将 Cookie 转发到源会降低缓存能力。对于将请求转发到 Amazon S3 源的缓存行为，请对 Forward Cookies (转发 Cookie) 选择 None (无)。

有关将 Cookie 转发到源的更多信息，请转到 CloudFront 如何转发、缓存及记录 Cookie (p. 61)。

Cookie 白名单（仅限 Amazon EC2 和其他自定义源）

如果您选择 Forward Cookies (转发 Cookie) 列表中的 Whitelist (白名单)，请在 Whitelist Cookies (Cookie 白名单) 字段中，输入您希望 CloudFront 为该缓存行为转发到您的原始服务器的 Cookie 名称。在新行中输入每个 Cookie 的名称。

默认情况下，如果您选择将白名单中的 Cookie 转发到您的源，则可配置每个缓存行为，以向您的原始服务器最多转发 10 个 Cookie 名称。要申请更高限额，请转到 https://aws.amazon.com/support/createCase?type=service_limit_increase&serviceLimitIncreaseType=cloudfront-distributions。

转发查询字符串

如果您的原始服务器基于 URL 中的查询字符串返回不同版本的对象，请单击 Yes (是)。如果您的源不管查询字符串而返回相同版本的对象，请单击 No (否)。这增加了 CloudFront 可从缓存提供请求的可能性，从而提高了性能并降低了源的负载。有关查询字符串的更多信息，请参阅 CloudFront 如何转发、缓存及记录查询字符串参数 (p. 59)。

限制查看者访问（使用签名 URL）

如果您希望与该缓存行为的 `PathPattern` 匹配的对象请求使用公共 URL，请单击 No (否)。

如果您希望与该缓存行为的 `PathPattern` 匹配的对象请求使用签名 URL，请单击 Yes (是)。然后指定您希望用于创建签名 URL 的 AWS 账户；这些账户被称为可信签署人。

有关可信签署人的更多信息，请参阅 指定可创建签名 URL 的 AWS 账户（可信签署人）(p. 99)。

可信签署人

选择您希望用作该缓存行为的可信签署人的 AWS 账户：

- Self (本人)：使用您目前登录 AWS Management Console 的账户作为可信签署人。如果您目前作为 IAM 用户登录，关联的 AWS 账户将作为可信签署人添加。
- Specify Accounts (指定账户)：在 AWS Account Numbers (AWS 账号) 字段中输入可信签署人的账号。

要创建签名 URL，AWS 账户必须至少具有一个有效的 CloudFront 密钥对。

Caution

如果您要更新您已经用于分发内容的分配，请在准备好开始为对象生成签名 URL 后仅添加可信签署人。在您将可信签署人添加到分配后，用户必须使用签名 URL 访问与该缓存行为的 `PathPattern` 匹配的对象。

AWS 账号

如果您要使用当前账户之外或非当前账户的 AWS 账户创建签名 URL，请在该字段中的每一行输入一个 AWS 账号。请注意以下几点：

- 您指定的账户必须至少具有一个有效的 CloudFront 密钥对。有关更多信息，请参阅 为可信签署人创建 CloudFront 密钥对 (p. 99)。
- 您不能为 IAM 用户创建 CloudFront 密钥对，因此，您不能使用 IAM 用户作为可信签署人。
- 有关如何获取账户的 AWS 账号的信息，请参阅 *Amazon Web Services General Reference* 中的我如何获取安全证书。
- 如果您输入当前账户的账号，CloudFront 会自动选中 Self (本人) 复选框，并从 AWS Account Numbers (AWS 账号) 列表中删除该账号。

分配的详细信息

以下值适用于整个分配。

价格级别

与您想要为 CloudFront 服务支付的最高价对应的价格级别。默认情况下，CloudFront 从所有 CloudFront 区域的节点提供您的对象。

有关价格级别以及您选择的价格级别如何影响 CloudFront 的分配处理性能的更多信息，请参阅 选择 CloudFront 分配的价格级别 (p. 54)。有关 CloudFront 定价的信息，包括价格级别与 CloudFront 区域的对应关系，请转到 Amazon CloudFront 定价。

备用域名（别名记录）

可选。指定您想用于对象 URL 的一个或多个域名，来代替您创建分配时 CloudFront 指派的域名。例如，如果您希望对象的 URL：

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

像这样：

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

而不是这样：

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

为 `www.example.com` 添加别名记录。

Important

如果您将 `www.example.com` 的别名记录添加到您的分配，还需要使用 DNS 服务创建（或更新）别名记录，以便将对 `www.example.com` 的查询路由至 `d111111abcdef8.cloudfront.net`。您必须具有权限，才能使用 DNS 服务提供商为域创建别名记录 (CNAME)。通常，这意味着您拥有该域，但您还可以为域的拥有者开发应用程序。

默认情况下，您最多可为每个分配创建 10 个别名记录。要申请更高限额，请转到 https://aws.amazon.com/support/createCase?type=service_limit_increase&serviceLimitIncreaseType=cloudfront-distributions。

有关备用域名的更多信息，请参阅 使用备用域名（别名记录）(p. 52)。有关 CloudFront URL 的更多信息，请参阅 CloudFront 对象的 URL 格式 (p. 57)。

SSL 证书

如果您希望查看者使用 HTTPS 访问您的对象，请选择适用的 SSL 证书：

- 默认 CloudFront 证书 (*.cloudfront.net)：如果您想要在您的对象的 URL 中使用 CloudFront 域名，例如 `https://d111111abcdef8.cloudfront.net/image1.jpg`，请选择该选项。

- IAM 证书存储中的证书名称：如果您想要在您的对象的 URL 中使用自己的域名，例如 `https://example.com/image1.jpg`，请选择以前上传到 IAM 证书存储中的适用证书。有关更多信息，请参阅 使用备用域名和 HTTPS (p. 136)。

默认根对象

可选。当查看者请求分配的根 URL (`http://www.example.com/`) 而不是分配中的对象 (`http://www.example.com/product-description.html`) 时，您希望 CloudFront 从您的源（例如，`index.html`）中请求的对象。指定一个默认根对象，以避免公开分配的内容。

名称的长度上限是 255 个字符。该名称可包含以下任意字符：

- A-Z，a-z
- 0-9
- _ - .* $ / ~ " '
- &（传递和返回为 `&`）

当您指定默认根对象时，请仅输入对象名称，例如，`index.html`。请不要在对象名称前添加 `/`。

有关更多信息，请参阅 指定默认根对象（仅 Web 分配）(p. 74)。

日志记录

您是否希望 CloudFront 记录关于每个对象请求的信息并将日志文件存储在 Amazon S3 存储桶中。您可以随时启用或禁用日志记录。启用日志记录无需额外费用，但在 Amazon S3 存储桶中存储和访问文件会产生常规的 Amazon S3 费用。您可以随时删除日志。有关 CloudFront 访问日志的更多信息，请参阅 访问日志 (p. 145)。

日志存储桶

对 Logging (日志记录) 选择 On (开) 时，您希望 CloudFront 用来存储访问日志的 Amazon S3 存储桶，例如 `myawslogbucket.s3.amazonaws.com`。如果您启用日志记录，CloudFront 会记录关于每个最终用户的对象请求的信息，并将文件存储在指定的 Amazon S3 存储桶中。您可以随时启用或禁用日志记录。有关 CloudFront 访问日志的更多信息，请参阅 访问日志 (p. 145)。

日志前缀

可选。对 Logging (日志记录) 选择 On (开) 时，指定您希望 CloudFront 添加到该分配的访问日志文件名前面的字符串（如果有），例如 `exampleprefix/`。尾斜杆 (/) 是可选的，但建议简化浏览您的日志文件。有关 CloudFront 访问日志的更多信息，请参阅 访问日志 (p. 145)。

Cookie 日志记录

如果您希望 CloudFront 将 Cookie 包含在访问日志中，请选择 On (开)。如果您选择将 Cookie 包含在日志中，CloudFront 会记录所有 Cookie，而不管您如何配置该分配的缓存行为：转发所有 Cookie，不转发 Cookie，或将指定的一组 Cookie 转发到源。

Amazon S3 不处理 Cookie，因此除非您的分配也包括 Amazon EC2 或其他自定义源，否则我们建议您对 Cookie Logging (Cookie 日志记录) 的值选择 Off (关)。

有关 Cookie 的更多信息，请转到 CloudFront 如何转发、缓存及记录 Cookie (p. 61)。

评论

可选。当您创建分配时，最多可包含 128 个字符的注释。您可以随时更新注释。

分配状态

指示您希望分配在部署后启用还是禁用：

- *Enabled (启用)* 意味着，分配完全部署后，您就可以部署使用分配的域名的链接，并且最终用户可检索内容。只要启用分配，CloudFront 便会接受并处理任何使用与该分配关联的域名的最终用户的内容请求。

  当您创建、修改或删除 CloudFront 分配时，需要时间将您的更改传播到 CloudFront 数据库。即刻发起的对分配相关信息的请求可能不会显示出该更改。传播通常在几分钟内完成，但高系统负载或网络分区可能会延长该时间。
- *Disabled (禁用)* 意味着，即使分配可能已经部署且准备就绪，但最终用户不能使用。只要禁用分配，CloudFront 便不接受任何使用与该分配关联的域名的最终用户请求。除非您将分配从禁用切换到启用（通过更新分配的配置），否则任何人都不能使用它。

您可以根据需要在禁用和启用之间频繁切换分配状态。请遵循更新分配配置的过程。有关更多信息，请参阅 列出、查看及更新 CloudFront 分配 (p. 55)。

自定义错误页面和错误缓存

您可以让 CloudFront 在您的 Amazon S3 或自定义源向 CloudFront 返回 HTTP 4xx 或 5xx 状态代码时向查看者返回一个对象（例如 HTML 文件）。您还可以指定从您的源发出的错误响应或自定义错误页面缓存在 CloudFront 边缘缓存中的时间。有关更多信息，请参阅 自定义错误响应 (p. 70)。

Note

以下值不包含在“Create Distribution (创建分配)”向导中，所以您只能在更新分配时配置自定义错误页面。

错误代码

您希望 CloudFront 返回自定义错误页面时所对应的 HTTP 状态代码。您可以将 CloudFront 配置为返回 CloudFront 缓存的针对无、部分或全部 HTTP 状态代码的自定义错误页面。

响应页面路径

您希望 CloudFront 在您的源返回您为 Error Code (错误代码) 指定的 HTTP 状态代码（例如 403）时，返回给查看者的自定义错误页面的路径（例如 `/4xx-errors/403-forbidden.html`）。如果您希望将您的对象和自定义错误页面存储在不同的位置，您的分配必须包含满足以下条件时的缓存行为：

- Path Pattern (路径模式) 的值与您的自定义错误消息的路径匹配。例如，您在 Amazon S3 存储桶的 `/4xx-errors` 目录下为 4xx 错误保存了自定义错误页面。您的分配必须包含其路径模式将对您的自定义错误页面的请求路由至该位置的缓存行为，例如 /4xx-errors/*。
- Origin (源) 值指定包含您的自定义错误页面的源的 Origin ID (源 ID) 值。

响应代码

您希望随自定义错误页面一起由 CloudFront 返回给查看者的 HTTP 状态代码。

错误缓存最小 TTL

您希望 CloudFront 缓存从您的原始服务器发出的错误响应的最短时间。

限制

如果您需要阻止选定国家/地区的用户访问您的内容，可以将您的 CloudFront 分配配置为允许指定国家/地区白名单中的用户访问您的内容或不允许指定国家/地区黑名单中的用户访问您的内容。有关更多信息，请参阅 限制您的内容的地理分配 (p. 42)。

Note

以下值不包含在“Create Distribution (创建分配)”向导中，所以您只能在更新分配时配置地理限制。

启用地理限制

您是否想要阻止选定国家/地区的用户访问您的内容。配置地理限制不收取额外费用。

限制类型

您希望如何指定可访问您的内容的用户所在的国家/地区：

- 白名单：该 Countries (国家/地区) 列表包含您*希望* 访问您的内容的用户所在的所有国家/地区。
- 黑名单：该 Countries (国家/地区) 列表包含您*不* 希望访问您的内容的用户所在的所有国家/地区。

国家/地区

您想要添加到白名单或黑名单中的国家/地区。要添加国家/地区，请在左侧列表中选择后单击 Add (添加)。请注意以下几点：

- 要添加多个连续的国家/地区，请选择第一个国家/地区，并按住 Shift 键选择最后一个国家/地区，然后单击 Add (添加)。
- 要添加多个不连续的国家/地区，请选择第一个国家/地区，并按住 Ctrl 键选择剩余的国家/地区，然后单击 Add (添加)。
- 要在左侧列表中查找国家/地区，请输入国家/地区全名的前几个字符。
- 如果您想要使用 CloudFront API 创建或更新分配，需要在每个国家/地区名称之前输入一个两个字母的代码值。我们使用 ISO 3166-1 - 国家/地区代码中的国际标准化组织国家/地区代码。

在您创建或更新 Web 分配时 CloudFront 显示在控制台中的值

当您创建新 Web 分配或更新现有分配时，CloudFront 将在 CloudFront 控制台中显示以下信息。

Note

有效可信签署人（具有有效 CloudFront 密钥对并可用于创建有效签名 URL 的 AWS 账户）目前在 CloudFront 控制台中不可见。

分配 ID（“General (常规)”选项卡）

当您使用 CloudFront API 对分配执行操作时，请使用分配 ID 指定您想要对其执行操作的分配，例如 `EDFDVBD6EXAMPLE`。您不能更改分配 ID。

分配状态（“General (常规)”选项卡）

下表中列出了分配的可能状态值。

值	描述
InProgress (进行中)	仍在创建或更新分配。
Deployed (已部署)	分配已创建或更新，并且更改已在整个 CloudFront 系统中完全传播。

Note

除了确保分配的状态是 Deployed (已部署)，您还必须启用分配，最终用户才能使用 CloudFront 访问您的内容。有关更多信息，请参阅 分配状态 (p. 39)。

上次修改时间（“General (常规)”选项卡）

上次修改分配的日期和时间，使用 ISO 8601 格式，例如 2012-05-19T19:37:58Z。有关更多信息，请转到 http://www.w3.org/TR/NOTE-datetime。

域名（“General (常规)”选项卡）

您在对象的链接中使用分配的域名。例如，您的分配的域名为 `d111111abcdef8.cloudfront.net`，`/images/image.jpg` 的链接将为 `http://d111111abcdef8.cloudfront.net/images/image.jpg`。您不能更改分配的 CloudFront 域名。有关您的对象链接的 CloudFront URL 的更多信息，请参阅 CloudFront 对象的 URL 格式 (p. 57)。

如果您指定了一个或多个备用域名（别名记录），则可对您的对象链接使用自己的域名，而不是使用 CloudFront 域名。有关别名记录的更多信息，请参阅 备用域名（别名记录）(p. 37)。

Note

CloudFront 域名是唯一的。您的分配的域名绝不会用于先前的分配，未来也绝不会再用于其他分配。

使用 Amazon EC2 和其他自定义源的要求和建议

请遵循在 CloudFront 中使用 Amazon EC2 实例和其他自定义源的原则。

- 在所有服务器上托管和提供相同的内容。
- 在所有服务器上记录 `X-Amz-Cf-Id` 标头项；CloudFront 调试时需要该信息。
- 限制对您的自定义源侦听的 HTTP 和 HTTPS 端口的访问请求。
- 在您的实施过程中，使所有服务器的时钟同步。
- 使用冗余服务器来处理故障。
- 有关请求和响应行为以及受支持的 HTTP 状态代码的信息，请参阅 请求和响应行为 (p. 80)。

如果您将 Amazon Elastic Compute Cloud 用于您的自定义源，我们建议您执行下列操作：

1. 使用亚马逊系统映像为 Web 服务器自动安装该软件。有关更多信息，请参阅 Amazon EC2 文档。

2. 使用 Elastic Load Balancing 负载均衡器处理多个 Amazon EC2 实例间的流量并将您的应用程序与 Amazon EC2 实例的更改隔离开。例如，您使用负载均衡器时，无需更改您的应用程序即可添加和删除 Amazon EC2 实例。有关更多信息，请参阅 Elastic Load Balancing 文档。
3. 当您创建 CloudFront 分配时，请为原始服务器的域名指定负载均衡器的 URL。有关更多信息，请参阅 创建 Web 分配 (p. 19)。

限制您的内容的地理分配

最终用户请求您的内容时，CloudFront 通常会提供请求的内容，而不考虑用户所在的位置。如果您需要阻止选定国家/地区的用户访问您的内容，可以配置 CloudFront Web 分配来执行以下操作之一：

- 仅当用户位于指定国家/地区的白名单中时才允许他们访问您的内容。
- 阻止位于指定国家/地区的黑名单中的用户访问您的内容。

例如，从一个因版权问题您无权分配内容的国家/地区发来一个请求，那么您可以阻止该请求；这又称为地理限制或地理阻止。

Note

CloudFront 使用第三方 GeoIP 数据库确定您的用户的位置。IP 地址和国家/地区之间映射的准确性因区域而异。根据最近的测试，我们的整体准确性为 99.8%。

以下是地理限制的工作机制：

1. 假设您仅有权在列支敦士登分配您的内容。您更新您的 CloudFront Web 分配并添加仅包含列支敦士登的白名单。（您也可以添加包含除列支敦士登以外的所有国家/地区的黑名单。）
2. 摩纳哥的一名用户请求您的内容，DNS 将该请求路由至意大利米兰中的 CloudFront 节点。
3. 米兰的节点查找您的分配并确定摩纳哥的这名用户不能下载您的内容。
4. CloudFront 向该用户返回 HTTP 状态代码 403（禁止）。

您可以选择性地将 CloudFront 配置为向该用户返回自定义错误消息，并可指定您希望针对所请求对象的错误响应缓存在 CloudFront 中的时间；默认值是五分钟。有关更多信息，请参阅 自定义错误响应 (p. 70)。

地理限制适用于整个分配。如果您需要对部分内容应用一个限制，而对另一部分内容应用不同限制，则必须创建单独的 CloudFront Web 分配。

如果您启用 CloudFront 访问日志记录，可使用 HTTP 状态代码 403 标识 CloudFront 拒绝的请求。但是，仅使用访问日志还不能区分 CloudFront 基于用户位置拒绝的请求与 CloudFront 因用户由于其他原因无权访问该对象而拒绝的请求。如果您有一项地理定位服务（例如 Digital Element 或 MaxMind），则可基于访问日志中 `c-ip`（客户端 IP）列的 IP 地址识别请求的位置。有关 CloudFront 访问日志的更多信息，请参阅 访问日志 (p. 145)。

Note

如果您需要限制您的内容在不按国家/地区界限划分的地理区域中的分配、您想要限制单个文件的分配或者想要将地理限制置入您的应用程序中，可将 CloudFront 与第三方服务组合使用。有关更多信息，请参阅教程 根据地理位置限制访问 CloudFront 分配中的文件（地理阻止）(p. 210)。

以下步骤介绍了如何使用 CloudFront 控制台将地理限制添加到现有的 Web 分配。有关如何使用控制台创建 Web 分配的信息，请参阅 创建 Web 分配 (p. 19)。

使用 CloudFront 控制台将地理限制添加到您的 CloudFront Web 分配

1. 登录 AWS 管理控制台，并通过以下网址打开 Amazon CloudFront 控制台：https://console.aws.amazon.com/cloudfront/。
2. 在 CloudFront 控制台的顶部窗格中，选择您想要更新的分配。

   Note

   顶部窗格将列出与您登录 CloudFront 控制台时所使用的 AWS 账户关联的所有分配。

3. 在 Distribution Settings (分配设置) 窗格中，单击 Restrictions (限制) 选项卡。
4. 单击 Edit (编辑)。
5. 输入适用的值。有关更多信息，请参阅 限制 (p. 40)。
6. 单击 Yes, Edit (是，编辑)。

使用 RTMP 分配

Topics

本节说明如何配置和管理 RTMP 分配。有关如何创建 RTMP 分配的更多信息，请参阅 创建 RTMP 分配 (p. 22)。

RTMP 分配的工作机制

要使用 CloudFront 流式处理媒体文件，您要向您的最终用户提供两种类型的文件：

- 您的媒体文件
- 媒体播放器，例如，JW Player、Flowplayer 或 Adobe Flash

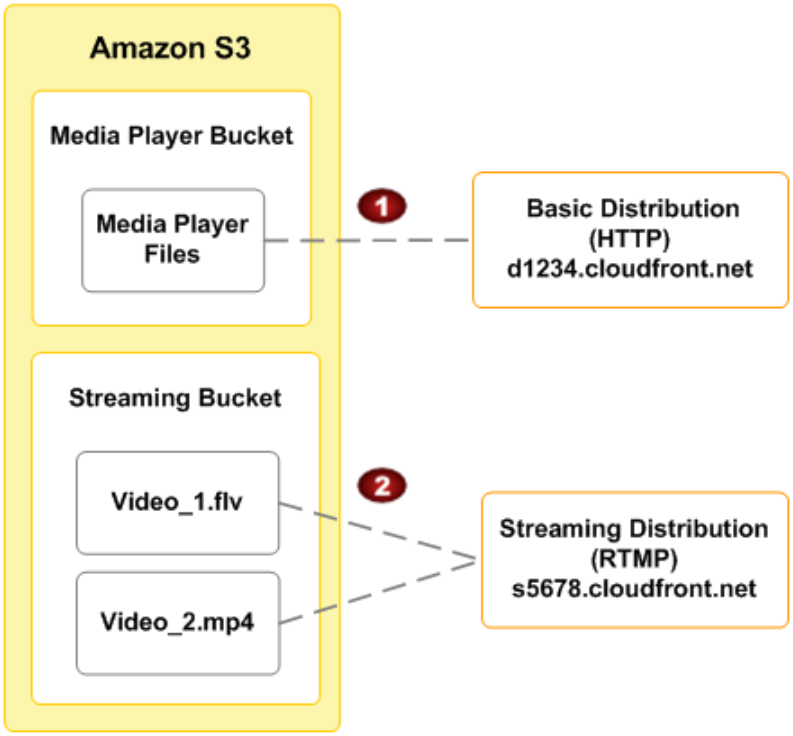
最终用户使用您提供给他们的媒体播放器查看您的媒体文件；他们不使用已经安装在其计算机或其他设备上的媒体播放器（如果有）。

当最终用户流式处理您的媒体文件时，媒体播放器在文件仍从 CloudFront 下载的过程中开始播放文件的内容。媒体文件不会存储在最终用户的本地系统上。

要使用 CloudFront 提供媒体播放器和媒体文件，您需要两种类型的分配：用于媒体播放器的 Web 分配和用于媒体文件的 RTMP 分配。Web 分配通过 HTTP 提供文件，而 RTMP 分配通过 RTMP（或 RTMP 的变体）流式处理媒体文件。

以下示例假设您的媒体文件和媒体播放器存储在不同的 Amazon S3 存储桶中，但这不是必需的 — 您可以将媒体文件和媒体播放器存储在相同的 Amazon S3 存储桶中。您还可以通过其他方式将媒体播放器提供给最终用户，例如使用 CloudFront 和自定义源。但是，媒体文件必须使用 Amazon S3 存储桶作为源。

在下图中，您的站点通过 `d1234.cloudfront.net` 域将媒体播放器的缓存副本提供给每个最终用户。然后，媒体播放器通过 `s5678.cloudfront.net` 域访问媒体文件的缓存副本。



1	您的媒体播放器存储桶存放媒体播放器，是常规 HTTP 分配的原始服务器。在该示例中，分配的域名是 `d1234.cloudfront.net`。（`d1234.cloudfront.net` 中的 d 表示这是 Web 分配。）
2	您的流媒体存储桶存放媒体文件，是 RTMP 分配的原始服务器。在该示例中，分配的域名是 `s5678.cloudfront.net`。（`s5678.cloudfront.net` 中的 s 表示这是 RTMP 分配。）

当您配置 CloudFront 以分配媒体文件时，CloudFront 使用 Adobe Flash Media Server 3.5 作为流服务器并使用 Adobe 实时消息协议 (RTMP) 流式处理您的媒体文件。CloudFront 通过 1935 和 80 端口接受 RTMP 请求。

CloudFront 支持 RTMP 协议的以下变体：

- RTMP —Adobe 实时消息协议
- RTMPT —通过 HTTP 打开隧道的 Adobe 流
- RTMPE —Adobe 加密流
- RTMPTE —通过 HTTP 打开隧道的 Adobe 加密流

有关 RTMP 以及 Adobe Flash Media Server 支持的文件格式的基本总结，请转到 Adobe 网站上的 Overview of Streaming with Flash Media Server 3（Flash Media Server 3 流处理概述）。此概述包括有关受支持的编解码器和容器的信息。

Internet 上提供的资源可帮助您确定用于 Flash 文件的比特率，例如，Adobe 网站上的 Flash video (FLV) bitrate calculator（Flash 视频 (FLV) 比特率计算器）。

CloudFront 支持 Adobe Flash Media Server 3.5 中与*动态流*相关的所有功能，这使您能够在播放过程中在不同质量的流之间进行切换。有关更多信息，请转到 Adobe 网站上的 Dynamic streaming in Flash Media Server 3.5: Part 1（Flash Media Server 3.5 动态流处理：第 1 部分）。

要提供流式处理的内容，您需要为最终用户提供媒体播放器。您可以使用 Adobe Flash 编写您自己的播放器。有关更多信息，请转到 http://www.adobe.com/products/flashplayer/。此外，您还可使用现有的播放器。有关更多信息，请参阅以下教程：

- 使用 CloudFront 和 Adobe Flash Player 的按需视频流 (p. 234)
- 使用 CloudFront 和 Flowplayer for Adobe Flash 的按需视频流 (p. 239)
- 使用 CloudFront 和 JW Player 的按需视频流 (p. 244)

使用 Amazon S3 存储桶作为 RTMP 分配的源

当您创建分配时，可指定 CloudFront 从何处获取分配到节点的文件。对于 RTMP 分配，您必须使用 Amazon S3 存储桶；自定义源不受支持。要使您的对象进入存储桶，可使用 Amazon S3 支持的任何方法，例如 Amazon S3 API 或第三方工具。您可以在您的存储桶中创建一个层次结构，就如使用任何其他 Amazon S3 存储桶一样。在存储桶中存储对象会产生常规的 Amazon S3 费用。有关 CloudFront 使用费的更多信息，请参阅 CloudFront 计费和使用 (p. 6)。

使用现有 Amazon S3 存储桶作为您的 CloudFront 原始服务器不会以任何方式更改存储桶；您仍然可以照常使用它来存储和访问 Amazon S3 对象（按常规 Amazon S3 价格）。

您可以将同一个 Amazon S3 存储桶同时用于 RTMP 和 Web 分配。

Note

在您创建 RTMP 分配后，就不能更改其原始服务器。如果您需要更改用于 RTMP 分配的 Amazon S3 存储桶，必须创建使用新存储桶的新分配，并更新您的链接或 DNS 记录以使用新分配的域名。然后，可删除原始分配。有关更多信息，请参阅 删除分配 (p. 56)。

当您指定您希望 CloudFront 从中获取对象的 Amazon S3 存储桶名称时，使用以下格式：

bucket-name.s3.amazonaws.com

请勿使用以下值指定存储桶的名称：

- Amazon S3 路径样式，s3.amazonaws.com/*bucket-name*
- Amazon S3 别名记录（如果有）

Important

要使您的存储桶能够与 CloudFront 一起使用，其名称必须符合 DNS 命名要求。有关详细信息，请参阅 *Amazon Simple Storage Service 开发者指南* 中的 Bucket Restrictions and Limitations。

为原始服务器创建多个 RTMP 分配

您一般为每个 Amazon S3 存储桶创建一个 RTMP 分配，但可以选择为同一存储桶创建多个 RTMP 分配。例如，您为一个 Amazon S3 存储桶创建了两个分配，则可使用任一分配引用一个媒体文件。在这种情况下，如果您的原始服务器中有一个名为 media.flv 的媒体文件，CloudFront 会使用每个分配，就像它们

各自引用了一个 media.flv 对象：可通过一个分配访问的一个 media.flv，和可通过另一个分配访问的另一个 media.flv。

您创建或更新 RTMP 分配时指定的值

要使用 CloudFront 流式处理媒体文件，请创建 RTMP 分配并指定以下值。

Topics

源域名（Amazon S3 存储桶）

您希望 CloudFront 从中获取此源的对象的 Amazon S3 存储桶的 DNS 域名，例如 myawsbucket.s3.amazonaws.com。在 CloudFront 控制台中，单击 Origin Domain Name (源域名) 字段，将显示一个列表，枚举与当前 AWS 账户关联的 Amazon S3 存储桶。要从不同的 AWS 账户使用存储桶，请按以下格式键入存储桶域名：

bucket-name.s3.amazonaws.com.

文件必须公开可读，除非您使用 CloudFront 原始访问标识确保 Amazon S3 中内容的安全。有关更多信息，请参阅 使用原始访问标识限制访问您的 Amazon S3 内容 (p. 95)。

Important

存储桶名称必须符合 DNS 命名要求。有关详细信息，请参阅 *Amazon Simple Storage Service 开发者指南* 中的 Bucket Restrictions and Limitations。

当您更改 CloudFront 从中获取当前源的对象的存储桶时，CloudFront 会立即开始将更改复制到 CloudFront 节点。在给定节点中更新分配配置前，CloudFront 会继续向以前的 Amazon S3 存储桶转发请求。该节点中的分配配置一经更新，CloudFront 就开始向新 Amazon S3 存储桶转发请求。

更改存储桶并不需要 CloudFront 用新源中的对象重新填充边缘缓存。只要您应用程序中的查看者请求没有更改，CloudFront 将继续提供边缘缓存中已有的对象，直至每个对象的 TTL 过期或很少被请求的对象被淘汰。

有关更多信息，请参阅 使用 Amazon S3 存储桶作为 RTMP 分配的源 (p. 45)。

限制存储桶访问（仅限 Amazon S3）

如果您希望要求最终用户仅使用 CloudFront URL（而非使用 Amazon S3 URL）访问 Amazon S3 存储桶中的对象，请单击 Yes (是)。然后指定适用的值。

如果您希望最终用户能够使用 CloudFront URL 或 Amazon S3 URL 访问对象，请单击 No (否)。

有关更多信息，请参阅 使用原始访问标识限制访问您的 Amazon S3 内容 (p. 95)。

原始访问标识（仅限 Amazon S3）

如果您对 Restrict Bucket Access (限制存储桶访问) 选择了 Yes (是)，请选择创建新原始访问标识还是使用与您的 AWS 账户关联的现有原始访问标识。如果您已经有原始访问标识，我们建议您重用该标识以便简化维护。有关原始访问标识的更多信息，请参阅 使用原始访问标识限制访问您的 Amazon S3 内容 (p. 95)。

新标识的注释（仅限 Amazon S3）

如果您对 Origin Access Identity (原始访问标识) 选择了 Create a New Identity (创建新标识)，请输入标识新原始访问标识的注释。当您创建该分配时，CloudFront 将创建该原始访问标识。

您的标识（仅限 Amazon S3）

如果您对 Origin Access Identity (原始访问标识) 选择了 Use an Existing Identity (使用现有标识)，请选择您想要使用的原始访问标识。您不能使用与另一 AWS 账户关联的原始访问标识。

授予对存储桶的读取权限（仅限 Amazon S3）

如果您希望 CloudFront 自动向原始访问标识授予读取您的 Amazon S3 存储桶中的对象的权限，请单击 Yes, Update Bucket Policy (是，更新存储桶策略)。

Important

如果您单击 Yes, Update Bucket Policy (是，更新存储桶策略)，CloudFront 会更新存储桶策略，以向指定的原始访问标识授予读取您的存储桶中对象的权限。然而，CloudFront 并不删除存储桶策略中的现有权限或对个别对象的权限。如果用户目前具有使用 Amazon S3 URL 访问您的存储桶中对象的权限，在 CloudFront 更新您的存储桶策略后，他们仍然具有该权限。要查看或更改现有存储桶策略以及对存储桶中对象的现有权限，请使用 Amazon S3 提供的方法。有关更多信息，请参阅 向原始访问标识授予读取 Amazon S3 存储桶中对象的权限 (p. 97)。

如果您希望手动更新权限，例如您希望更新对象的 ACL 而非更新存储桶权限，请单击 No, I will Update Permissions (否，我将更新权限)。

价格级别

与您想要为 CloudFront 服务支付的最高价对应的价格级别。默认情况下，CloudFront 从所有 CloudFront 区域的节点提供您的对象。

有关价格级别以及您选择的价格级别如何影响 CloudFront 的分配处理性能的更多信息，请参阅 选择 CloudFront 分配的价格级别 (p. 54)。有关 CloudFront 定价的信息，包括价格级别与 CloudFront 区域的对应关系，请转到 Amazon CloudFront 定价。

备用域名（别名记录）

可选。您可以将一个或多个别名记录别名与分配关联起来，以便在对象的 URL 中使用您自己的域名（例如，example.com），而非您创建分配时使用 CloudFront 指派的域名。有关更多信息，请参阅 使用备用域名（别名记录） (p. 52)。

日志记录

您是否希望 CloudFront 记录关于每个对象请求的信息并将日志文件存储在 Amazon S3 存储桶中。您可以随时启用或禁用日志记录。启用日志记录无需额外费用，但在 Amazon S3 存储桶中存储和访问文件会产生常规的 Amazon S3 费用。您可以随时删除日志。有关 CloudFront 访问日志的更多信息，请参阅 访问日志 (p. 145)。

日志存储桶

对 Logging (日志记录) 选择 On (开) 时，您希望 CloudFront 用来存储访问日志的 Amazon S3 存储桶，例如 `myawslogbucket.s3.amazonaws.com`。如果您启用日志记录，CloudFront 会记录关于每个最终用户的对象请求的信息，并将文件存储在指定的 Amazon S3 存储桶中。您可以随时启用或禁用日志记录。有关 CloudFront 访问日志的更多信息，请参阅 访问日志 (p. 145)。

日志前缀

可选。对 Logging (日志记录) 选择 On (开) 时，指定您希望 CloudFront 添加到该分配的访问日志文件名前面的字符串（如果有），例如 `exampleprefix/`。尾斜杆 (/) 是可选的，但建议简化浏览您的日志文件。有关 CloudFront 访问日志的更多信息，请参阅 访问日志 (p. 145)。

评论

可选。当您创建分配时，最多可包含 128 个字符的注释。您可以随时更新注释。

分配状态

当您创建分配时，必须指定希望分配在创建后启用还是禁用：

- *Enabled (启用)* 意味着，分配完全部署后，您就可以部署使用分配的域名的链接，并且最终用户可检索内容。只要启用分配，CloudFront 便会接受并处理任何使用与该分配关联的域名的最终用户的内容请求。

  当您创建、修改或删除 CloudFront 分配时，需要时间将您的更改传播到 CloudFront 数据库。即刻发起的对分配相关信息的请求可能不会显示出该更改。传播通常在几分钟内完成，但高系统负载或网络分区可能会延长该时间。
- *Disabled (禁用)* 意味着，即使分配可能已经部署且准备就绪，但最终用户不能使用。当禁用分配时，CloudFront 不会接受任何使用与该分配关联的域名的最终用户请求。除非您将分配从禁用切换到启用（通过更新分配的配置），否则任何人都不能使用它。

您可以根据需要在禁用和启用之间频繁切换分配状态。有关更新分配配置的信息，请参阅 列出、查看及更新 CloudFront 分配 (p. 55)。

限制查看者访问（使用签名 URL）

如果您希望对由该分配提供的对象的请求使用公共 URL，请单击 No (否)。如果您希望请求使用签名 URL，请单击 Yes (是)。然后指定您希望用于创建签名 URL 的 AWS 账户；这些账户被称为可信签署人。

有关可信签署人的更多信息，请参阅 指定可创建签名 URL 的 AWS 账户（可信签署人） (p. 99)。

可信签署人

选择您希望用作该分配的可信签署人的 AWS 账户：

- Self (本人)：使用您目前登录 AWS Management Console 的账户作为可信签署人。如果您目前作为 IAM 用户登录，关联的 AWS 账户将作为可信签署人添加。
- Specify Accounts (指定账户)：在 AWS Account Numbers (AWS 账号) 字段中输入可信签署人的账号。

要创建签名 URL，AWS 账户必须至少具有一个有效的 CloudFront 密钥对。

Caution

如果您要更新您已经用于分发内容的分配，请在准备好开始为对象生成签名 URL 后仅添加可信签署人。在您将可信签署人添加到分配中后，用户必须使用签名 URL 访问该分配提供的对象。

AWS 账号

如果您要使用当前账户之外或非当前账户的 AWS 账户创建签名 URL，请在该字段中的每一行输入一个 AWS 账号。请注意以下几点：

- 您指定的账户必须至少具有一个有效的 CloudFront 密钥对。有关更多信息，请参阅 为可信签署人创建 CloudFront 密钥对 (p. 99)。
- 您不能为 IAM 用户创建 CloudFront 密钥对，因此，您不能使用 IAM 用户作为可信签署人。
- 有关如何获取账户的 AWS 账号的信息，请参阅 *Amazon Web Services General Reference* 中的我如何获取安全证书。
- 如果您输入当前账户的账号，CloudFront 会自动选中 Self (本人) 复选框，并从 AWS Account Numbers (AWS 账号) 列表中删除该账号。

在您创建或更新 RTMP 分配时 CloudFront 显示在控制台中的值

当您创建新 RTMP 分配或更新现有分配时，CloudFront 将在 CloudFront 控制台中显示以下信息。

Note

有效可信签署人（具有有效 CloudFront 密钥对并可用于创建有效签名 URL 的 AWS 账户）目前在 CloudFront 控制台中不可见。

分配 ID

当您使用 CloudFront API 对分配执行操作时，请使用分配 ID 指定您想要对其执行操作的分配，例如 `EDFDVBD6EXAMPLE`。您不能更改分配 ID。

状态

下表中列出了分配的可能状态值。

值	描述
InProgress (进行中)	仍在创建或更新分配。

值	描述
Deployed (已部署)	分配已创建或更新，并且更改已在整个 CloudFront 系统中完全传播。

除了确保分配的状态是 Deployed (已部署)，您还必须启用分配，最终用户才能使用 CloudFront 访问您的内容。有关更多信息，请参阅 分配状态 (p. 48)。

上一次修改

上次修改分配的日期和时间，使用 ISO 8601 格式，例如 2012-05-19T19:37:58Z。有关更多信息，请转到 http://www.w3.org/TR/NOTE-datetime。

域名

除非您使用备用域名（别名记录），否则在您对象的链接中使用分配的域名。例如，如果分配的域名为 `d111111abcdef8.cloudfront.net`，示例 `/images/image.jpg` 文件的链接将为 `http://d111111abcdef8.cloudfront.net/images/image.jpg`。您不能更改分配的 CloudFront 域名。有关您的对象链接的 CloudFront URL 的更多信息，请参阅 CloudFront 对象的 URL 格式 (p. 57)。

如果您指定了一个或多个备用域名（别名记录），则可对您的对象链接使用自己的域名，而不是使用 CloudFront 域名。有关别名记录的更多信息，请参阅 备用域名（别名记录）(p. 37)。

Note

CloudFront 域名是唯一的。您的分配的域名绝不会用于先前的分配，未来也绝不会再用于其他分配。

配置媒体播放器

要播放媒体文件，您必须使用文件的正确路径配置媒体播放器。您如何配置媒体取决于您使用哪个媒体播放器以及如何使用它。

当您配置媒体播放器时，您指定的媒体文件路径必须包含紧随域名之后的字符 `cfx/st`，例如：

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

Note

CloudFront 遵循 Adobe 的 FMS 命名要求。有关如何指定流，不同的播放器具有各自的规则。上述示例是 JW Player。请检查您的播放器的说明文件。例如，Adobe Flash Media Server 不允许 `.flv` 扩展名出现在播放路径上。许多播放器都为您删除 `.flv` 扩展名。

您的媒体播放器可能要求路径与文件名分开。例如，在 JW Player 向导中，您要指定 `streamer` 和 `file` 变量：

- streamer —`rtmp://s5c39gqb8ow64r.cloudfront.net/cfx/st`（无尾斜杆）
- file — `mediafile.flv`

如果您已经在存储桶的目录中存储了媒体文件（例如，`videos/mediafile.flv`），那么 JW Player 的变量将为：

- streamer —`rtmp://s5c39gqb8ow64r.cloudfront.net/cfx/st`（无尾斜杆）
- file — `videos/mediafile.flv`

要使用 JW Player 向导，请转到 JW Player 网站上的 Setup Wizard（设置向导）页面。

MPEG 文件

要提供 MP3 音频文件或 H.264/MPEG-4 视频文件，您可能需要用 `mp3:` 或 `mp4:` 作为文件名的前缀。某些媒体播放器可配置为自动添加前缀。媒体播放器还可能要求您指定无文件扩展名的文件名（例如，`magicvideo`，而非 `magicvideo.mp4`）。

使用 Crossdomain.xml 限制访问

Adobe Flash Media Server `crossdomain.xml` 文件指定哪些域能访问特定域内的媒体文件。CloudFront 提供一个默认文件，允许所有域访问您的 RTMP 分配中的媒体文件，而您无法更改该行为。如果您将一个限制性更强的 `crossdomain.xml` 文件包含到您的 Amazon S3 存储桶中，CloudFront 将忽略它。

关于 RTMP 分配的错误代码

下表列出了 CloudFront 可发送到您的媒体播放器的错误代码。这些错误是随 `Event.info.application.message` 或 `Event.info.description` 返回的部分字符串。

错误	描述
`DistributionNotFound`	未找到分配。
`DistributionTypeMismatch`	分配不是 RTMP 分配。
`InvalidInstance`	实例无效。
`InvalidURI`	URI 无效。

对 RTMP 分配进行故障排除

如果媒体文件播放出现问题，请检查以下项目。

要检查的项目	描述
媒体播放器文件和媒体文件分别有各自的分配	媒体播放器必须由常规的 HTTP 分配（例如，域名 d111111abcdef8.cloudfront.net）提供，而媒体文件必须由 RTMP 分配（例如，域名 s5c39gqb8ow64r.cloudfront.net）提供。请确保您不会对两者使用同一分配。
文件路径中的 `/cfx/st`	请确认文件路径中包含 `/cfx/st`。您不需要将 `/cfx/st` 包含在 Amazon S3 存储桶中对象的路径中。有关更多信息，请参阅 配置媒体播放器 (p. 50)。
MPEG-4 文件名	有些媒体播放器要求在文件名前加 `mp4:`。有些也可能要求您排除 `.mp4` 扩展名。有关更多信息，请参阅 MPEG 文件 (p. 51)。
您防火墙上的端口 1935	Adobe Flash Media Server 为 RTMP 使用端口 1935。请确保您的防火墙已打开该端口。如果没有，返回的典型消息是“Unable to play video (无法播放视频)”。您还可以切换到 RTMPT 以使用端口 80 通过 HTTP 打开隧道。

要检查的项目	描述
Adobe Flash Player 消息	默认情况下，如果 Adobe Flash Player 试图播放的视频文件缺失，则不会显示消息。相反，它会等待文件出现。您可能希望改变此行为，以给予最终用户更好的体验。 为使播放器在视频缺失时发送消息，请使用 `play("vid",0,-1)`，而非 `play("vid")`。

使用备用域名（别名记录）

Topics

在 CloudFront 中，备用域名也称为别名记录 (CNAME)，让您可以将自己的域名（例如 `www.example.com`）用于指向您的对象的链接，而不是使用 CloudFront 指派给分配的域名。Web 和 RTMP 分配均支持备用域名。

当您创建分配时，CloudFront 会返回用于分配的域名，例如：

```
d111111abcdef8.cloudfront.net
```

当您使用 CloudFront 域名用于您的对象时，名为 `/images/image.jpg` 的对象的 URL 是：

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

如果您想使用自己的域名（例如 `www.example.com`），而非 CloudFront 指派给您的分配的 `cloudfront.net` 域名，可以为 `www.example.com` 向您的分配中添加一个备用域名。这样您就可以使用 `/images/image.jpg` 的以下 URL：

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

在备用域名中使用 * 通配符

当您添加备用域名时，可在域名开头使用 * 通配符，而不是单独指定子域。例如，备用域名为 `*.example.com`，您就可以在您的对象 URL 中使用以 example.com 结尾的任何域名，例如 `www.example.com`、`product-name.example.com` 和 `marketing.product-name.example.com`。无论域名如何，对象的名称相同，例如：

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

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

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

备用域名必须以星号和点 (`*.`) 开头。您*不能*使用通配符替代部分子域名，例如 `*domain.example.com`；也不能在域名中间替代子域名，例如 `subdomain.*.example.com`。

只要在同一 CloudFront 分配中，通配符备用域名可以与另一个备用域名重叠。例如，您可以同时使用 `www.example.com` 和 `*.example.com` 作为备用域名，但它们必须在同一分配中。

备用域名的使用限制

请注意备用域名的以下使用限制：

- 默认情况下，您最多可为每个分配添加 10 个备用域名。要申请更高限额，请转到 https://aws.amazon.com/support/createCase?type=service_limit_increase&serviceLimitIncreaseType=cloudfront-distributions。
- 您必须具有权限，才能使用 DNS 服务提供商为域创建别名记录 (CNAME)。通常，这意味着您拥有该域，但您还可以为域的拥有者开发应用程序。
- 如果备用域名已存在于一个 CloudFront 分配中，则即使您的 AWS 账户拥有这个分配，也不能将该备用域名添加到另一个 CloudFront 分配中。
- DNS 协议不允许您为 DNS 命名空间的顶端节点（也称为主域顶点）创建别名记录。例如，如果您注册 DNS 名称 `example.com`，则主域顶点为 `example.com`。您不能为 `example.com` 创建别名记录，但可为 `www.example.com`、`newproduct.example.com` 等创建别名记录。

  如果您使用 Route 53 作为您的 DNS 服务，可以创建别名资源记录集，而非别名记录。有了别名资源记录集，您就不用付款进行 Route 53 查询了。此外，您可以在主域顶点 (example.com) 为域名创建别名资源记录集。有关更多信息，请转到 *Amazon Route 53 开发人员指南* 中的将查询路由至 Amazon CloudFront 分配。
- 如果您希望查看者使用 HTTPS 和备用域名，需要进行其他配置。有关更多信息，请参阅 使用备用域名和 HTTPS (p. 136)。

添加备用域名

以下任务列表描述了如何使用 CloudFront 控制台将备用域名添加到分配，以便您在您的链接中使用自己的域名，而非与您的分配关联的 CloudFront 域名。

您还可使用 CloudFront API 更新您的分配：

- 要更新 Web 分配，请使用 `PUT Distribution Config` API 操作。有关更多信息，请转到 *Amazon CloudFront API 参考* 中的 PUT 分配配置。
- 要更新 RTMP 分配，请使用 `PUT Streaming Distribution Config` API 操作。有关更多信息，请转到 *Amazon CloudFront API 参考* 中的 PUT 流分配配置。

使用 CloudFront 控制台添加备用域名的过程

1. 登录 AWS 管理控制台，并通过以下网址打开 Amazon CloudFront 控制台：https://console.aws.amazon.com/cloudfront/。
2. 在 CloudFront 控制台中，使用以下步骤更新您的分配，以将您的域名作为备用域名包含在 Alternate Domain Names (CNAMEs) (备用域名 (别名记录)) 字段中。

   此外，如果您希望查看者使用 HTTPS 和您的备用域名，请在 SSL Certificate (SSL 证书) 列表中选择合适的 SSL 证书。有关使用 HTTPS 和备用域名的更多信息，请参阅 使用备用域名和 HTTPS (p. 136)。

   a. 在 CloudFront 控制台的顶部窗格中，选择您想要更新的分配，然后单击 Distribution Settings (分配设置)。
   b. 在 General (常规) 选项卡上，单击 Edit (编辑)。
   c. 在 Alternate Domain Names (CNAMEs) (备用域名 (别名记录)) 字段中添加相应的备用域名。用逗号隔开多个域名，或将每个域名放在一个新行中。
   d. 如果适用，在 SSL Certificate (SSL 证书) 列表中，选择一个 SSL 证书。
   e. 单击 Yes, Edit (是，编辑)。

3. 在 CloudFront 控制台中您的分配的 General (常规) 选项卡上，确认分配的状态已更改为 Deployed (已部署)。如果您试图在部署分配更新之前使用备用域名，则在后续步骤中创建的链接可能无法正常工作。
4. 使用 DNS 服务提供商提供的方法，将别名记录资源记录集添加到域的托管区域。这个新的别名记录资源记录集会将 DNS 查询从您的域（例如 www.example.com）重定向到您的分配的 CloudFront 域名（例如 d111111abcdef8.cloudfront.net）。有关更多信息，请参阅 DNS 服务提供商提供的文档。

   如果您使用 Amazon Route 53 作为您的 DNS 服务，可以创建别名资源记录集，而非别名记录。有了别名资源记录集，您就不用付款进行 Route 53 查询了。此外，您可以在主域顶点 (example.com) 为域名创建别名资源记录集，而 DNS 不允许在此创建别名记录。有关更多信息，请转到 *Amazon Route 53 开发人员指南* 中的将查询路由至 Amazon CloudFront 分配。

   Important

   如果您的域名已经具有现成的别名记录，请更新此资源记录集或将其替换为指向您的分配的 CloudFront 域名的新记录集。

   此外，确认您的别名记录资源记录集指向您的分配的域名，而不是您的原始服务器之一。

5. 使用 dig 或类似工具，确认您在第 4 步创建的别名记录资源记录集指向您的分配的域名。有关 dig 的更多信息，请转到 http://www.kloth.net/services/dig.php。

   以下示例显示了 images.example.com 域上的 dig 请求以及响应的相关部分。

```
[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.
...
...
```

   Answer Section 一行显示了将对 images.example.com 的查询路由至 CloudFront 分配域名 d111111abcdef8.cloudfront.net 的别名记录资源记录集。如果 CNAME 右侧的名称是您的 CloudFront 分配的域名，则表示别名记录资源记录集配置正确。如果是任何其他值，例如 Amazon S3 存储桶的域名，则表示别名记录资源记录集配置错误。在这种情况下，请返回第 4 步，纠正别名记录，以指向您的分配的域名。
6. 通过创建 URL 中使用您的域名（而不是您的分配的 CloudFront 域名）的一些测试链接，来测试备用域名。
7. 在您的应用程序中，更改您的对象的链接，以使用您的备用域名，而不是您的 CloudFront 分配的域名。

选择 CloudFront 分配的价格级别

CloudFront 拥有遍布世界各地的节点。我们每个节点的费用各不相同，因此，向您收取的价格视 CloudFront 用来处理您的请求的节点而定。

CloudFront 节点按地理区域分组，而且我们已经将区域划分为不同的价格级别。默认的价格级别包括所有区域。另一个价格级别包括大部分区域（美国；欧洲；香港、韩国和新加坡；日本；以及印度地区），但排除最昂贵的区域。第三个价格级别仅包括最便宜的区域（美国和欧洲地区）。

默认情况下，CloudFront 仅基于性能响应对您的对象的请求：由对于该查看者延迟最低的节点提供对象。如果您愿意接受让某些地理区域的查看者承受较高的延迟，以换取低成本，可以选择不包括所有 CloudFront 区域的价格级别。虽然 CloudFront 将只从该价格级别的节点提供您的对象，但它仍然从您选定的价格级别的节点中选择延迟最低的节点来提供内容。但是，与您的内容从所有 CloudFront 节点提供相比，您的部分查看者（特别是不在您的价格级别里的地理区域中的那部分）可能会承受更高的延迟。例如，如果您选择只包括美国和欧洲的价格级别，与您选择包括澳大利亚和亚洲的价格级别相比，澳大利亚和亚洲的查看者可能会感受到更高的延迟。

如果您选择不包括所有节点的价格级别，CloudFront 可能仍然会偶尔从不包含在您的价格级别中的区域内的节点处理对您的内容的请求。当这种情况发生时，不会按照提供您的对象的较昂贵区域的费率向您收费，而是按照您选定价格级别中最便宜区域的费率收费。

您可以在创建 CloudFront 分配时选择价格级别，或使用 CloudFront 控制台或 CloudFront API 更新现有的分配。要查找有关使用 CloudFront 控制台或 API 创建或更新 Web 或 RTMP 分配的相应主题，请参阅对分配的操作 (p. 27)。

有关 CloudFront 定价和价格级别的更多信息，请转到 Amazon CloudFront 定价。

列出、查看及更新 CloudFront 分配

您可以使用 CloudFront 控制台列出与您的 AWS 账户关联的 CloudFront 分配、查看分配的设置并更新大部分设置。

当您将更改保存到您的分配配置时，CloudFront 开始向所有节点传播更改。您的配置在节点中更新前，CloudFront 会基于先前配置继续从该位置提供您的内容。您的配置在节点中更新后，CloudFront 会立即基于新配置从该位置提供您的内容。

您所做的更改不会即刻传播到每个节点；传播到所有节点可能需要 15 分钟。传播完成后，您的分配的状态将从 InProgress (进行中) 更改为 Deployed (已部署)。CloudFront 正在将您的更改传播到节点之时，我们不能确定指定节点是基于先前配置还是新配置提供您的内容。

使用 CloudFront 控制台列出、查看及更新 CloudFront 分配

1. 登录 AWS 管理控制台，并通过以下网址打开 Amazon CloudFront 控制台：https://console.aws.amazon.com/cloudfront/。
2. 在 CloudFront 控制台的顶部窗格中，选择您想要查看或更新的分配。

   Note

   顶部窗格将列出与您登录 CloudFront 控制台时所使用的 AWS 账户关联的所有分配。

3. 要查看或编辑 RTMP 分配设置，请跳至第 4 步。

   要查看或编辑 Web 分配的设置，请执行以下步骤。

   a. 在 Distribution Settings (分配设置) 窗格中，单击您想要更改的设置所对应的选项卡：General (常规)、Origins (源) 或 Behaviors (行为)。
   b. 对于常规设置，请单击 Edit (编辑)。

      对于源或缓存行为，单击源或缓存行为，然后单击 Edit (编辑)。
   c. 输入或更新适用的值。有关字段的信息，请参阅以下主题：

- 常规设置：分配的详细信息 (p. 37)
- 源设置：源设置 (p. 31)
- 缓存行为设置：缓存行为设置 (p. 33)

d. 单击 Yes, Edit (是，编辑)。

4. 编辑或查看 RTMP 分配的设置：

a. 在 Distribution Details (分配的详细信息) 窗格中，单击 Edit (编辑)。
b. 输入或更新适用的值。有关字段的信息，请参阅 您创建或更新 RTMP 分配时指定的值 (p. 46)。
c. 单击 Yes, Edit (是，编辑)。

删除分配

如果您不希望再使用分配，请遵循以下程序使用 CloudFront 控制台将其删除。

您还可使用 CloudFront API 删除分配：

- 要删除 Web 分配，请使用 `DELETE Distribution` API 操作。有关更多信息，请转到 *Amazon CloudFront API 参考* 中的 DELETE 分配。
- 要删除 RTMP 分配，请使用 `DELETE Streaming Distribution` API 操作。有关更多信息，请转到 *Amazon CloudFront API 参考* 中的 DELETE 流分配。

Note

CloudFront 可让您为一个 AWS 账户最多创建总共 100 个 Web 和 RTMP 分配。

使用 CloudFront 控制台删除 CloudFront 分配

1. 登录 AWS 管理控制台，并通过以下网址打开 Amazon CloudFront 控制台：https://console.aws.amazon.com/cloudfront/。
2. 在 CloudFront 控制台的右侧窗格中，找到您想要删除的分配。
3. 如果 State (状态) 列的值为 Disabled (禁用)，则跳到第 7 步。

   如果 State (状态) 的值是 Enabled (启用)，Status (状态) 的值是 Deployed (已部署)，则继续执行第 4 步以禁用分配，然后再删除它。

   如果 State (状态) 的值是 Enabled (启用)，Status (状态) 的值是 InProgress (进行中)，请等待 Status (状态) 更改为 Deployed (已部署)。然后继续执行第 4 步以禁用分配，然后再删除它。
4. 在 CloudFront 控制台的右侧窗格中，选中您想要删除的分配对应的复选框。
5. 单击 Disabled (禁用) 以禁用分配，然后单击 Yes, Disable (是，禁用) 进行确认。然后单击 Close (关闭)。
6. State (状态) 列的值将立即更改为 Disabled (禁用)。请等待 Status (状态) 列的值更改为 Deployed (已部署)。
7. 选中您想要删除的分配对应的复选框。
8. 单击 Delete (删除)，然后单击 Yes, Delete (是，删除) 进行确认。然后单击 Close (关闭)。

使用对象

Topics

本节说明如何使用 CloudFront 中的对象。

CloudFront 对象的 URL 格式

Topics

在创建分配之后，您会收到与该分配相关联的 CloudFront 域名。您在创建对象链接时需要使用此域名。如果您想使用其他域名（例如，`www.example.com`），可以添加一个别名记录 (CNAME) 别名。有关更多信息，请参阅 使用备用域名（别名记录）(p. 52)。

当您创建 URL 来允许最终用户访问您的 CloudFront 分配中的对象时，创建的 URL 可以是公共 URL 也可以是签名 URL：

公共 URL - 允许用户访问以下对象：

- 不受任何限制的对象。

- 最终用户必须通过 CloudFront 进行访问，但访问时不需要签名 URL 的 Amazon S3 存储桶中的对象。这些对象不能使用 Amazon S3 URL 进行访问。

签名 URL - 如果您将某个缓存行为配置为需要签名 URL，那么在访问该缓存行为指定的对象时，就需要使用签名 URL。请注意，如果针对某个对象（例如，`image.jpg`）的请求与两个或更多缓存行为的路径模式相匹配，则 CloudFront 将根据分配中第一个列出的缓存行为来对请求进行处理。如果第一个缓存行为不需要签名 URL，而第二个缓存行为需要签名 URL，则最终用户将可以在无需签名 URL 的情况下访问 `image.jpg`。

有关缓存行为的更多信息，包括路径模式，请参阅缓存行为设置 (p. 33)。有关签名 URL 的更多信息，请参阅通过 CloudFront 提供私有内容。 (p. 91)。

Amazon S3 中对象的公共 URL 格式

Amazon S3 存储桶中对象的公共 URL 使用以下格式：

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

Important

如果分配提供流内容，则文件路径中需要更多字符。有关更多信息，请参阅配置媒体播放器 (p. 50)。

例如，假设您有一个名为 `mybucket` 的 Amazon S3 存储桶。该存储桶包含名为 `/images/image.jpg` 的公共可读对象。

您创建一个 CloudFront 分配并指定 `mybucket.s3.amazonaws.com` 作为此分配的原始服务器。CloudFront 将返回 `d111111abcdef8.cloudfront.net` 作为该分配的域名并返回 `EDFDVBD6EXAMPLE` 作为分配 ID。

您提供给最终用户用于访问此示例中对象的 URL 是：

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

对于 Web 分配，如果您要在多个 Amazon S3 存储桶中存储您的内容，则 URL 的格式是相同的 — URL 不包含任何有关 Amazon S3 存储桶的信息。要将请求路由至适用的存储桶，您需要为每个存储桶创建一个源，并创建一个或多个缓存行为用于将请求路由至每个源。缓存行为中的路径模式指定了将哪些请求路由至与该缓存行为关联的源（Amazon S3 存储桶）。有关 CloudFront 分配中的源和缓存行为设置的更多信息，请参阅您创建或更新 Web 分配时指定的值 (p. 29)。

有关 Amazon S3 存储桶的名称和路径的更多信息，请参阅*Amazon Simple Storage Service 开发者指南*中的存储桶的虚拟托管部分。

任何时候，只要有最终用户访问该对象，CloudFront 都会从适当的节点提供它。如果对象不在节点上，则 CloudFront 将转到与 `EDFDVBD6EXAMPLE` 分配 (`mybucket.s3.amazonaws.com`) 相关联的原始服务器，并获取节点对象的副本以提供给最终用户。

自定义源中对象的公共 URL 格式

自定义源中对象的公共 URL 格式很像 Amazon S3 中对象的公共 URL：

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

如果您的对象位于原始服务器上的一个文件夹中，则 CloudFront URL 必须包含该文件夹的名称。例如，如果 `image.jpg` 位于 `images` 文件夹中，则 URL 为：

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

CloudFront 使用公共 URL 中的对象路径和名称从您创建分配时指定的域中获取对象。例如，如果您的自定义源的域为 `example.com` 且对象路径和名称为 `/images/image.jpg`，则 CloudFront 将从以下位置获取对象：

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

如果您将内容存储在多个自定义源中，则URL的格式是相同的—URL不包含任何有关自定义源的信息。要将请求路由至适用的自定义源，您需要在您的分配中为每个自定义源添加一个源，并创建一个或多个缓存行为用于将请求路由至每个源。缓存行为中的路径模式指定了要将哪些请求路由至与该缓存行为关联的源。有关 CloudFront 分配中的源和缓存行为设置的更多信息，请参阅您创建或更新 Web 分配时指定的值 (p. 29)。

公共 URL 如何影响目录的失效

如果您使用允许最终用户访问目录的 CloudFront URL，我们建议您始终使用相同的 URL 格式，统一带末尾的斜杠 (/) 或统一不带，例如：

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

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

浏览器和其他 Web 应用程序会将这两种格式解析为同一目录。但是，CloudFront 是完全按照您指定的公共 URL 格式来存储它们，而您如果希望使目录失效，就需要指定完全相同的目录，包括或不包括斜杠。如果您没有关于指定目录的标准，您将需要使带斜杠和不带斜杠的目录都失效，以确保 CloudFront 从节点删除该目录。如果您已达到了月免费失效的限额，将需要为这两种失效付费，即使只存在一个目录。

签名 URL 的格式

签名 URL 允许最终用户访问配置为提供私有内容的分配中的对象。此类 URL 包括限制访问缓存对象的额外信息。有关签名 URL 格式的信息，请参阅通过 CloudFront 提供私有内容。 (p. 91)。

CloudFront 如何处理 HTTP 和 HTTPS 请求

默认情况下，对于 Amazon S3 源，CloudFront 接受通过 HTTP 和 HTTPS 协议发出的针对 CloudFront 分配中对象的请求。然后，CloudFront 使用发出请求时所用的协议将请求传递到您的 Amazon S3 存储桶或自定义源。

对于自定义源，您可以在创建分配时指定 CloudFront 访问您的源的方式：仅限 HTTP，或者匹配查看器使用的协议。有关 CloudFront 如何处理针对自定义源的 HTTP 和 HTTPS 请求的更多信息，请参阅协议 (p. 86)。

有关如何对您的 Web 分配进行限制以使最终用户只能使用 HTTPS 访问对象的信息，请参阅使用 HTTPS 连接访问您的对象 (p. 134)。（此选项不适用于使用 RTMP 协议的 RTMP 分配。）

Note

HTTPS 请求的费用高于 HTTP 请求的费用。有关账单费率的更多信息，请转到 CloudFront 定价计划。

CloudFront 如何转发、缓存及记录查询字符串参数

对于 Web 分配，您可以选择您是否希望 CloudFront 将查询字符串参数转发到您的源。对于 RTMP 分配，您无法将 CloudFront 配置为向您的源转发查询字符串参数。

对于上述两种类型的分配，如果您启用日志记录，则 CloudFront 会记录完整的 URL，包括查询字符串参数。对于 Web 分配，无论您是否配置为让 CloudFront 转发查询字符串都是如此。有关 CloudFront 日志记录的更多信息，请参阅访问日志 (p. 145)。

有关更多信息，请参阅相关主题：

- 查询字符串参数和 Web 分配 (p. 60)
- 查询字符串参数和 RTMP 分配 (p. 61)

查询字符串参数和 Web 分配

对于 Web 分配，您可以指定您是否希望 CloudFront 在向您的源转发请求时包括查询字符串。例如，您可以指定您是否希望 CloudFront 转发以下 URL 中的 *?parameter1=a* 部分：

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

如果您将 CloudFront 配置为向您的源转发查询字符串，则 CloudFront 在缓存对象时将包括 URL 的查询字符串部分。例如，以下查询字符串导致 CloudFront 缓存三个对象。即使您的源始终返回相同的 `image.jpg` 也是如此，无论查询字符串是什么：

- `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`

如果您的源根据查询字符串返回某个对象（例如，`/images/image.jpg`）的不同版本，请在 CloudFront 控制台中为 Forward Query Strings (转发查询字符串) 选择 Yes (是) 或在您使用 CloudFront API 时将 `DistributionConfig` 复杂类型中 `QueryString` 元素的值指定为 `true`。

如果您的源无论查询字符串是什么都返回相同版本的对象，请选择 No (否) 或 `false`。这增加了 CloudFront 可从缓存提供请求的可能性，从而提高了性能并降低了源的负载。

查询字符串中参数的顺序很重要。如果您将 CloudFront 配置为向您的源转发查询字符串，则以下查询字符串会导致 CloudFront 缓存两个对象：

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

查询字符串中的大小写也很重要。如果您将 CloudFront 配置为向您的源转发查询字符串，则以下查询字符串会导致 CloudFront 缓存两个对象：

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

如果您使用签名 URL 限制对您内容的访问（如果您向您的分配中添加了可信签署人），则 CloudFront 将在向您的源转发 URL 剩余部分之前删除以下查询字符串参数：

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

这意味着，如果您要使用签名 URL 且将 CloudFront 配置为向您的源转发查询字符串参数，则您无法将自己的查询字符串参数命名为 `Expires`、`Key-Pair-Id`、`Policy` 或 `Signature`。

查询字符串参数和 RTMP 分配

对于 RTMP 分配，当 CloudFront 从原始服务器请求对象时，它将删除任何查询字符串参数。例如，如果 CloudFront 收到以下请求而 `media.flv` 尚未在 CloudFront 缓存中存在：

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

它将向您的原始服务器发送以下 URL：

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

CloudFront 如何转发、缓存及记录 Cookie

对于 HTTP 和 HTTPS Web 分配，您可选择您是否希望 CloudFront 向您的源转发 Cookie。对于 RTMP 分配，您不能将 CloudFront 配置为处理 Cookie。

Important

Amazon S3 和一些 HTTP 服务器不处理 Cookie。切勿将 CloudFront 缓存行为配置为向不能处理 Cookie 的源转发 Cookie，否则您会对缓存能力产生不利影响，进而对性能产生不利影响。

Cookie 和 Web 分配

对于 Web 分配，您可选择您是否希望 CloudFront 向您的源转发 Cookie。如果是，您可选择您是希望 CloudFront 转发所有 Cookie 还是希望它仅转发您指定的 Cookie。如果您选择只转发您指定的 Cookie，您最多可为分配中的每个缓存行为指定 10 个 Cookie。要请求增大每个缓存行为 10 个 Cookie 的默认限额，请转到 https://aws.amazon.com/support/createCase?type=service_limit_increase&serviceLimitIncreaseType=cloudfront-distributions。

当您将 CloudFront 配置为向您的源转发 Cookie 时：

- CloudFront 在向您的源转发请求时，将包括 Cookie 的名值对而非 Cookie 的属性（例如，路径或过期时间）。例如，在以下 `Cookie` 标题中，CloudFront 转发 Cookie 名值对 `country=ata` 而不转发 Cookie 属性 `$Path=/`：

  ```
  Cookie: country=ata; $Path=/;
  ```
- 如果您将 CloudFront 配置为只转发指定列表中的 Cookie，则 CloudFront 在向您的源转发请求之前将删除任何不在此列表中的 Cookie。
- 不支持 `If-Modified-Since` 和 `If-None-Match` 条件请求。
- CloudFront 在向您的源转发请求之前会按 Cookie 名称的自然顺序对 Cookie 进行排序。

当您的源将 Cookie 返回到 CloudFront 时：

- 如果您将 CloudFront 配置为在发生指定的缓存行为时将所有 Cookie 转发到您的源，则 CloudFront 将缓存您的源返回的响应以及所有 Cookie 和 Cookie 属性。CloudFront 还会将对象以及所有 Cookie 和 Cookie 属性返回到查看器。
- 如果您将 CloudFront 配置为在发生指定的缓存行为时仅向您的源转发指定的 Cookie，则 CloudFront 将缓存响应、指定的 Cookie 以及关联的 Cookie 属性。CloudFront 还会将仅包括指定的 Cookie 和 Cookie 属性的对象返回到查看器。
- 如果您不希望 CloudFront 缓存 Cookie 和 Cookie 属性，请配置您的原始服务器以将以下标题添加到 CloudFront 的响应中：

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

如果您将 CloudFront 配置为向您的源转发 Cookie，则 CloudFront 将使用查看器包括在请求中的 Cookie 来唯一识别缓存中的对象。例如，如果对象 `/images/image1.jpg` 的三个请求包含三个不同的名值对，则 CloudFront 将缓存相同的对象三次，每个名值对缓存一次。即使您的源忽略请求中的 Cookie 值，并始终向 CloudFront 返回相同的 `image1.jpg` 对象也是如此。结果就是，CloudFront 向您的源转发针对同一个对象的多个请求，这会降低性能并增加原始服务器的负载。如果您的原始服务器不根据给定 Cookie 的值改变其响应，我们建议您不要将 CloudFront 配置为向您的源转发此 Cookie。

Cookie 的名称和值都区分大小写。例如，如果同一对象的两个 Cookie 除大小写之外完全相同，则 CloudFront 将缓存此对象两次。

如果您将 CloudFront 配置为记录请求和 Cookie，则 CloudFront 将记录所有 Cookie 以及所有 Cookie 属性，即使您将 CloudFront 配置为不向您的源转发 Cookie 或将 CloudFront 配置为仅转发指定列表中的 Cookie 也是如此。有关 CloudFront 日志记录的更多信息，请参阅访问日志 (p. 145)。

有关使用 CloudFront 控制台更新分配以使 CloudFront 向源转发 Cookie 的信息，请参阅列出、查看及更新 CloudFront 分配 (p. 55)。有关使用 CloudFront API 更新分配的信息，请转到 *Amazon CloudFront API 参考* 中的 PUT Distribution Config。

Cookie 和 RTMP 分配

对于 RTMP 分配，当 CloudFront 从原始服务器请求对象时，它将在向您的源转发此请求之前删除任何 Cookie。如果您的源随此对象返回任何 Cookie，则 CloudFront 将在向查看器返回该对象之前删除它们。对于 RTMP 分配，CloudFront 不在边缘缓存中缓存 Cookie。

对于 RTMP 分配，您无法将 CloudFront 配置为记录 Cookie。

在分配中添加、删除或替换对象

有关为分配添加对象的信息，请参阅 向 CloudFront 分配中添加对象 (p. 62)。

当您替换分配中的对象时，我们建议您使用版本控制的对象名称。有关更多信息，请参阅 使用版本控制的对象名称更新现有对象 (p. 62)。您还可用具有相同名称的对象替换对象。请参阅 使用相同对象名称更新现有对象 (p. 63)。不管您选择如何替换您分配中的对象，我们都建议您指定应从 CloudFront 缓存中删除对象的时间。有关更多信息，请参阅 指定对象在 CloudFront 边缘缓存中的保留时间（过期）(p. 63)。

如果您需要快速删除分配中的对象，可以使其失效。有关更多信息，请参阅 使对象失效（仅 Web 分配）(p. 66)。

向 CloudFront 分配中添加对象

当您向您的源添加对象时，请确保将其添加到分配的一个 Amazon S3 存储桶中，或者，对于自定义源，将其添加到指定域中的目录。

当您将对象添加到您的源并公开对象的 CloudFront 链接时，CloudFront 节点在收到最终用户对该对象的请求之前将不会从源获取对象。

CloudFront 服务器不确定它们提供的对象的 MIME 类型。当您向您的源上传对象时，应设置对象的 `Content-Type` 标题字段。

使用版本控制的对象名称更新现有对象

当您更新 CloudFront 分配中的现有对象时，我们建议您在对象名称或目录名称中包括某种版本标识符，从而使您可以更好地控制自己的内容。此标识符可能是日期时间戳、序列号、或区别同一对象的两种版本的其他方法。

例如，您可能不想将图像文件命名为 image.jpg，而是命名为 image_1.jpg。当您想要开始提供新版本的文件时，您会将新文件命名为 image_2.jpg，并且您会更新链接以指向 image_2.jpg。此外，您可将所有的图形放在 images_v1 目录中，且当您想要开始提供一个或多个图像的新版本时，您会创建新的 images_v2 目录，并且您会更新指向该目录的链接。借助版本控制，您不必在 CloudFront 开始提供对象新版本之前等待其过期，并且您不必为对象的失效支付费用。

即使您对对象进行版本控制，我们仍建议您设置过期日期。有关更多信息，请参阅 指定对象在 CloudFront 边缘缓存中的保留时间（过期）(p. 63)。

Note

指定版本控制的对象名称或目录名称与 Amazon S3 对象版本控制无关。

使用相同对象名称更新现有对象

虽然您可以更新 CloudFront 分配中的现有对象并使用相同的对象名称，但是我们并不建议您这么做。CloudFront 仅在请求对象时将其分配到节点，在您将新的或更新的对象放置到源中时不会执行分配。如果您使用具有相同名称的较新版本更新源中的现有对象，节点将不会从源中获取新版本，除非发生以下两种情况：

- 缓存中对象的旧版本过期。有关更多信息，请参阅 指定对象在 CloudFront 边缘缓存中的保留时间（过期）(p. 63)。
- 最终用户请求该节点上的对象。

如果您在替换对象时使用相同的名称，则您将无法控制 CloudFront 开始提供新文件的时间。默认情况下，CloudFront 在节点缓存对象 24 小时。（有关更多信息，请参阅 指定对象在 CloudFront 边缘缓存中的保留时间（过期）(p. 63)。）例如，如果您要替换整个网站中的所有对象：

- 不太受欢迎页面的对象可能不会出现在任何节点。在下次请求时将开始提供这些对象的新版本。
- 有些页面的对象可能存在于一些节点中而不在另外一些节点中，因此，最终用户将会看到不同的版本，具体取决于提供对象的节点。
- 不会长达 24 小时地提供最热门网页对象的新版本，这是因为 CloudFront 可能在您用新版本替换对象之前就已从这些页面获取了对象。

指定对象在 CloudFront 边缘缓存中的保留时间（过期）

Topics

您可控制对象在 CloudFront 向您的源转发另一请求之前在 CloudFront 缓存中的保留时间。减少持续时间让您可提供动态内容。增加持续时间意味着直接从边缘缓存提供对象的可能性增大，因此您的客户可以获得更好的性能。较长的持续时间还会减少源的负载。

Note

您还可以控制在 CloudFront 通过向您的源转发另一请求再次尝试获取请求的对象之前，错误（例如，错误 404，“not found (未找到)”）在 CloudFront 缓存中的保留时间。有关更多信息，请参阅 CloudFront 如何处理与缓存 HTTP 4xx 和 5xx 状态码 (p. 88)。

通常，CloudFront 在对象过期之前从节点提供对象。对象过期后，在节点下次获得最终用户对该对象的请求时，CloudFront 会将请求转发到原始服务器，以验证缓存中是否包含对象的最新版本：

- 如果 CloudFront 已有最新版本，则源将仅返回 304 状态代码（not modified (未修改)）。
- 如果 CloudFront 没有最新版本，则源将返回 200 状态代码 (OK) 和对象的最新版本。

如果节点中的对象未获得频繁的请求，CloudFront 可能会将该对象逐出 — 在对象过期日期前删除对象 — 以为更受欢迎的对象腾出空间。

默认情况下，每个对象在 24 小时后自动过期。要指定不同的过期时间，请配置您的源，以将 `Cache-Control max-age` 指令或 `Expires` 标题字段的值添加到每个对象：

- 通过 `Cache-Control max-age` 指令，您可以指定希望对象在 CloudFront 再次从原始服务器获取对象之前在缓存中保留的时间（按秒计）。对于 Web 分配和 RTMP 分配，CloudFront 支持的最短过期时间分别为 0 秒和 3600 秒。最大值为 in the year 2038。按以下格式指定值：

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

  例如，以下指令告诉 CloudFront 在缓存中将关联的对象保留 3600 秒（一小时）：

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

  如果您希望对象在 CloudFront 边缘缓存中保留的时间不同于在浏览器缓存中的保留时间，可以将 `Cache-Control max-age` 和 `Cache-Control s-maxage` 指令一起使用。有关更多信息，请参阅指定 CloudFront 为 Web 分配缓存对象的最短时间 (p. 65)。
- 通过 `Expires` 标题字段，您可以使用 RFC 2616, Hypertext Transfer Protocol -- HTTP/1.1 Section 3.3.1, Full Date 中指定的格式指定过期日期和时间，例如：

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

  Important

  在过了 `Expires` 标题中指定的过期日期和时间后，每次节点收到对该对象的请求时，CloudFront 都会再次从原始服务器中获取对象。

我们建议您使用 `Cache-Control max-age` 指令代替 `Expires` 标题字段来控制对象缓存。如果您同时指定 `Cache-Control max-age` 和 `Expires` 的值，则 CloudFront 将仅使用 `max-age` 的值。

您不能使用最终用户 `GET` 请求中的 HTTP `Cache-Control` 或 `Pragma` 标题字段来迫使 CloudFront 返回到对象的原始服务器。CloudFront 将忽略最终用户的这些标题字段。

有关 `Cache-Control` 和 `Expires` 标题字段的更多信息，请参阅《*RFC 2616，超文本传输协议 – HTTP/1.1*》中的以下章节：

- 第 14.9 节：缓存控制
- 第 14.21 节：过期

有关如何使用适用于 PHP 的 AWS 开发工具包添加 `Cache-Control` 和 `Expires` 标题字段的示例，请参阅*Amazon Simple Storage Service 开发者指南*中的使用适用于 PHP 的 AWS 开发工具包上传对象。一些第三方工具也可添加这些字段。

指定 CloudFront 为 Web 分配缓存对象的最短时间

对于 Web 分配，如果您要为对象添加 Cache-Control 或 Expires 标题，还可以指定 CloudFront 在向源转发另一请求之前在缓存中保留对象的最短时间。以下显示了对象标题和最小 TTL 如何控制对象在 CloudFront 缓存中的保留时长：

	最小 TTL = 0（默认）	最小 TTL > 0
源添加**Cache-Control max-age**指令到对象	对象缓存时长为 Cache-Control max-age 指令的值。	对象缓存时长为 Cache-Control max-age 指令的值或 CloudFront Minimum TTL (最小 TTL) 的值，以两者中较大的值为准。
源不添加 **Cache-Control max-age** 指令	对象缓存 24 小时。	对象缓存时长为 24 小时或 CloudFront Minimum TTL (最小 TTL)，以两者中较大的值为准。
源向对象中添加 **Cache-Control max-age** 和 **Cache-Control s-maxage** 指令	对象在 CloudFront 边缘缓存中的缓存时长为 Cache-Control s-maxage 指令的值。它们在浏览器缓存中的缓存时长为 Cache-Control max-age 指令的值。	对象在 CloudFront 边缘缓存中的缓存时长为 Cache-Control s-maxage 指令的值或 CloudFront Minimum TTL (最小 TTL) 的值，以两者中较大的值为准。它们在浏览器缓存中的缓存时长为 Cache-Control max-age 指令的值。
源添加 **Expires** 标题	对象缓存至 Expires 标题中的日期。在过了此日期后，CloudFront 会将每个请求转发到源。	对象缓存至 Expires 标题中的日期。在过了此日期后，对象缓存时长为 CloudFront Minimum TTL (最小 TTL) 的值。
源将 **Cache-Control no-cache**、**no-store** 和/或 **private** 指令添加到对象	CloudFront 尊重标题。	对象缓存时长为 Minimum TTL (最小 TTL)。

有关如何使用 CloudFront 控制台更改 Web 分配的设置的信息，请参阅 列出、查看及更新 CloudFront 分配 (p. 55)。有关如何使用 CloudFront API 更改 Web 分配的设置的信息，请参阅 PUT 配置。

指定 CloudFront 为 RTMP 分配缓存对象的最短时间

对于 RTMP 分配，默认情况下，CloudFront 在边缘缓存保留对象 24 小时。在 CloudFront 向源转发另一请求之前，您可以为对象添加 Cache-Control 或 Expires 标题，以将缓存持续时间缩减至 1 小时（3600 秒）。如果您指定一个较低的值，则 CloudFront 将使用 3600 秒。

使用 Amazon S3 控制台将标题添加到您的对象

Note

使用 Amazon S3 控制台，您一次只能为一个对象添加标题；而使用一些第三方工具，您一次可以为多个 Amazon S3 对象添加标题。有关支持 Amazon S3 的第三方工具的更多信息，请通过网络搜索 **AWS S3** 。

要使用 Amazon S3 控制台为 Amazon S3 对象添加 **`Cache-Control`** 或 **`Expires`** 标题字段，请按以下步骤操作

1. 登录 AWS 管理控制台，并通过以下网址打开 Amazon S3 控制台：https://console.aws.amazon.com/s3/。
2. 在 Amazon S3 控制台中，在“存储桶”窗格里，单击包含该文件的存储桶名称。
3. 在 Objects and Folders (对象和文件夹) 窗格中，选择您想添加标题字段的第一个对象。
4. 在 Objects and Folders (对象和文件夹) 窗格的顶部，单击 Actions (操作)，然后单击 Properties (属性)。
5. 在 Properties (属性) 窗格中，单击 Metadata (元数据) 选项卡。
6. 在“Metadata (元数据)”选项卡上，单击 Add More Metadata (添加更多元数据)。
7. 在 Key (键) 列表中，单击 Cache-Control (缓存控制) 或 Expires (过期)，如适用。
8. 在 Value (值) 字段中，输入适用的值：
   - 对于 `Cache-Control` 字段，输入：

     `max-age=`*`number of seconds that you want objects to stay in a CloudFront edge cache`*
   - 对于 Expires (过期) 字段，输入 HTML 格式的日期和时间。
9. 单击 Save (保存)。
10. 如果您想要为其他对象添加标题字段，请在 Objects and Folders (对象和文件夹) 窗格中单击下一个对象的名称，然后重复第 6 步到第 9 步。

使对象失效（仅 Web 分配）

Topics

如果您需要在对象过期前从 CloudFront 边缘服务器缓存中删除对象，可以执行以下操作之一：

- 使对象失效。最终用户下次请求对象时，CloudFront 将返回源以获取对象的最新版本。
- 使用对象版本控制提供具有不同名称的对象的不同版本。有关更多信息，请参阅 使用版本控制的对象名称更新现有对象 (p. 62)。

Important

您只能使 Web 分配提供的对象失效。您不能使 RTMP 分配提供的对象失效。

您每月可免费使指定数量的对象失效。您需要为您使其失效的超过该限额的每个对象支付费用。例如，要使目录及目录中的所有文件失效，您必须分别使目录和每个文件失效。如果您需要使大量文件失效，则创建新分配，然后更改对象路线以关联到新分配可能更容易、更便宜。有关使用失效的费用的更多信息，请参阅 支付对象失效费用 (p. 70)。

在使对象失效和使用版本控制的对象名称之间进行选择

要控制分配提供的对象版本，您可以使对象失效或为其指定版本控制的文件名。如果您将要频繁地更新您的对象，出于以下原因，我们建议您主要使用对象版本控制：

- 即使当最终用户具有本地缓存版本或在企业缓存代理后缓存的版本时，您仍可以使用版本控制来控制请求返回哪个对象。如果您使对象失效，最终用户可能会继续看到旧版本，直至它从这些缓存中过期。
- 文件名包含在 CloudFront 访问日志中，因此，版本控制使分析对象更改的结果变得更加容易。
- 版本控制提供一种将不同版本的对象提供给不同最终用户的方式。
- 版本控制简化了对象修订版之间的前滚和后滚。
- 版本控制花费更少。您仍需为 CloudFront 付费以将对象的新版本传输到节点，但您不必为使对象失效而为每个文件支付费用。

有关对象版本控制的更多信息，请参阅 使用版本控制的对象名称更新现有对象 (p. 62)。

确定使哪些对象失效

如果您要使所有 CloudFront 边缘缓存中的所有对象失效而您的用户不必访问源上的每个对象，则您可以确定查看器已从 CloudFront 请求的对象并仅使这些对象失效。要确定查看器已请求的对象，请启用 CloudFront 访问日志记录。有关访问日志的更多信息，请参阅 访问日志 (p. 145)。

使对象失效并显示有关失效的信息

您可使用 CloudFront 控制台或 CloudFront API 操作来创建并运行失效、显示您先前提交的失效列表以及显示有关单个失效的详细信息。您还可复制现有的失效、编辑对象路径列表以及运行已编辑的失效。

请参阅适用主题：

- 使用 CloudFront 控制台使对象失效 (p. 67)
- 使用 CloudFront 控制台复制、编辑及重新运行现有失效 (p. 68)
- 使用 CloudFront 控制台列出失效 (p. 69)
- 使用 CloudFront 控制台显示有关失效的信息 (p. 69)
- 使用 CloudFront API 使对象失效并显示有关失效的信息 (p. 69)

使用 CloudFront 控制台使对象失效

您可创建任何数量的失效，但您一次只能使每个分配的三个失效处于进行中。每个请求最多可包含 1000 个要失效的对象。如果您超过这些限制，则 CloudFront 控制台将显示错误消息。要确定当前有多少个进行中的失效，请查看 Invalidations (失效) 选项卡上的 Status (状态) 列。

要使用 CloudFront 控制台使对象失效，请按以下步骤操作

1. 登录 AWS 管理控制台，并通过以下网址打开 Amazon CloudFront 控制台：https://console.aws.amazon.com/cloudfront/。
2. 单击您想要使其对象失效的分配。
3. 单击 Distribution Settings (分配设置)。
4. 单击 Invalidations (失效) 选项卡。
5. 单击 Create Invalidation (创建失效)。
6. 输入您要使其失效的对象路径。请注意以下几点：
   - 您只能使与 Web 分配关联的对象失效。

- 您必须明确使您希望 CloudFront 停止服务的每个对象和每个目录失效。您不能使用通配符使对象组失效，而且，您不能通过指定目录路径使目录中的所有对象失效。
- 路径是相对于分配的。前倒斜杠是可选的。例如，要使 `http://d111111abcdef8.cloudfront.net/images/image2.jpg` 处的对象失效，您可指定：

  `/images/image2.jpg`

  或者

  `images/image2.jpg`
- 如果您将 CloudFront 配置为向您的源转发查询字符串，则在使对象失效时，必须包括查询字符串，例如：
  - `images/image.jpg?parameter1=a`
  - `images/image.jpg?parameter1=b`

  如果客户端请求包括同一对象的五个不同的查询字符串，则您必须使该对象失效五次，每个查询字符串一次。有关更多信息，请参阅 CloudFront 如何转发、缓存及记录查询字符串参数 (p. 59)。要确定已使用哪些查询字符串，可以启用 CloudFront 日志记录。有关更多信息，请参阅 访问日志 (p. 145)。
- 如果您正在使用签名 URL，可以通过在问号 (?) 前仅包括部分 URL 来使对象失效。
- 如果对象在目录中且您尚未使用标准化方法来指定目录 — 带或不带尾斜杠 — 我们建议您同时使带和不带尾斜杠的目录失效，例如，`images` 和 `images/`。有关更多信息，请参阅 公共 URL 如何影响目录的失效 (p. 59)。
- 您最多可指定 1000 个对象。
- 路径的长度上限是 4000 个字符。
- 要使默认根对象失效，请以您为任何其他对象指定路径相同的方式指定路径。
- 如果路径包括非 ASCII 字符或 RFC 1783 (http://www.ietf.org/rfc/rfc1738.txt) 中定义的不安全字符，则需要对这些字符进行 URL 编码。请勿对路径中的任何其他字符进行 URL 编码，否则，CloudFront 将不会使已更新对象的旧版本失效。

7. 单击 Invalidate (失效)。

使用 CloudFront 控制台复制、编辑及重新运行现有失效

您可复制您先前创建的失效、更新对象路径列表以及运行已更新的失效。您不能复制现有失效、更新对象路径以及在不运行已更新失效的情况下保存它。

Important

如果您复制仍在进行中的失效、更新对象路径列表以及运行已更新的失效，CloudFront 将不会停止或删除您复制的失效。如果任何对象路径出现在源和副本中，CloudFront 将尝试使对象失效两次，且两次失效均计算在您每月免费失效的最大数目中。如果您已经达到了免费失效的最大数目，将会向您收取每个对象两次失效的费用。有关更多信息，请参阅 失效限制 (p. 70)。

要使用 CloudFront 控制台复制、编辑及重新运行现有失效，请按以下步骤操作

1. 登录 AWS 管理控制台，并通过以下网址打开 Amazon CloudFront 控制台：https://console.aws.amazon.com/cloudfront/。
2. 单击包含您要复制的失效的分配。
3. 单击 Distribution Settings (分配设置)。
4. 单击 Invalidations (失效) 选项卡。
5. 单击您要复制的失效。

如果您不确定要复制哪一失效，可以单击失效，然后单击 Details (详细信息) 以显示该失效的详细信息。

6. 单击 Copy (复制)。
7. 更新对象路径列表，如适用。
8. 单击 Invalidate (失效)。

使用 CloudFront 控制台列出失效

使用此控制台，您可显示您已为分配创建和运行的最后 100 个失效的列表。如果您想获得超过 100 个失效的列表，请使用 GET 失效列表 API 操作。有关更多信息，请转到*Amazon CloudFront API 参考*中的 GET 失效列表。

要使用 CloudFront 控制台列出失效，请按以下步骤操作

1. 登录 AWS 管理控制台，并通过以下网址打开 Amazon CloudFront 控制台：https://console.aws.amazon.com/cloudfront/。
2. 单击您想为其显示失效列表的分配。
3. 单击 Distribution Settings (分配设置)。
4. 单击 Invalidations (失效) 选项卡。

使用 CloudFront 控制台显示有关失效的信息

您可显示失效的详细信息，包括分配 ID、失效 ID、失效状态、创建失效的日期和时间以及完整的对象路径列表。

要使用 CloudFront 控制台显示有关失效的信息，请按以下步骤操作

1. 登录 AWS 管理控制台，并通过以下网址打开 Amazon CloudFront 控制台：https://console.aws.amazon.com/cloudfront/。
2. 单击包含您想显示详细信息的失效的分配。
3. 单击 Distribution Settings (分配设置)。
4. 单击 Invalidations (失效) 选项卡。
5. 单击您想显示详细信息的失效。
6. 单击 Details (详细信息)。

使用 CloudFront API 使对象失效并显示有关失效的信息

有关使用 CloudFront API 使对象失效及显示失效相关信息的信息，请参阅*Amazon CloudFront API 参考*中的相关主题：

- 使对象失效：POST 失效
- 获得失效列表：GET 失效列表
- 获得有关具体失效的信息：GET 失效

用于使对象失效的第三方工具

除 CloudFront 提供的失效方法之外，还可以使用一些第三方工具来使对象失效。有关工具列表，请参阅使对象失效 (p. 250)。

失效限制

您可作出任何数量的失效请求，但您一次只能使每个分配的三个失效请求处于进行中。每个请求最多可包含 1000 个要失效的对象。如果您超过这些限制，CloudFront 将返回错误消息。

Note

CloudFront 通常需要 10 到 15 分钟来完成您的失效请求，具体取决于请求的大小。

支付对象失效费用

您每月请求的前 1000 个对象失效是免费的；您须为每月超过 1000 个的每个对象失效支付费用。该限制适用于您用一个 AWS 账户创建的所有分配的对象失效总数。例如，如果您使用 AWS 账户 `john@example.com` 创建三个分配，且每个分配在指定月份里有 600 个对象失效（总数为 1 800 个失效），则 AWS 在该月将向您收取 800 个对象失效的费用。有关失效定价的具体信息，请转到 Amazon CloudFront 定价。

Note

就失效定价而言，对象失效请求被定义为单个 `Path` 元素对象。有关 `Path` 元素的更多信息，请参阅 使对象失效并显示有关失效的信息 (p. 67)。

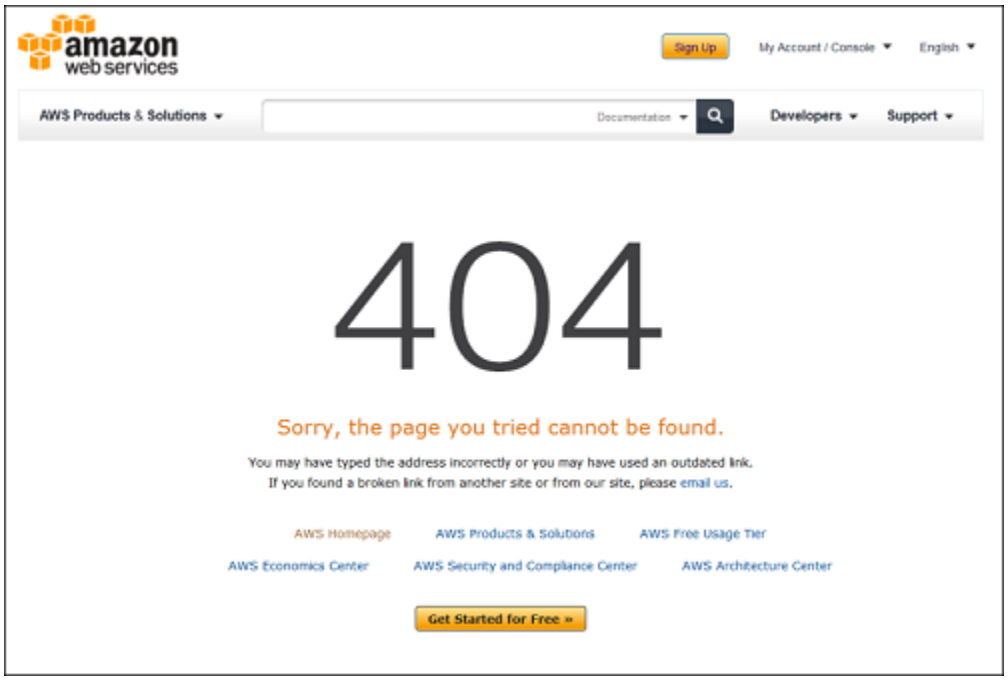
自定义错误响应

Topics

如果您通过 CloudFront 提供的对象出于某种原因不可用，您的 Web 服务器通常会返回 HTTP 状态代码到 CloudFront。例如，如果查看器指定了无效 URL，您的 Web 服务器将返回 404 状态代码至 CloudFront，然后 CloudFront 会将此状态代码返回至查看器。查看器显示简要且采用稀疏格式的默认消息，如下所示：

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

如果您更愿意显示自定义错误消息，可能与您网站其他部分使用相同的格式设置，则您可以让 CloudFront 向查看器返回包含您的自定义错误消息的对象（例如，html 文件）。



您可以为每个受支持的 HTTP 状态代码指定一个不同的对象，也可以对所有受支持的状态代码使用同一个对象。还可以选择为一些状态代码指定对象，而不为另外一些状态代码指定对象。

您通过 CloudFront 提供的对象出于各种原因可能不可用。可将这些原因归为以下两大类：

- 客户端错误表示请求出现问题。例如，具有指定名称的对象不可用，或用户不具有获取您 Amazon S3 存储桶中的对象所需的权限。当出现客户端错误时，源会向 CloudFront 返回一个 400 区间内的 HTTP 状态代码。
- 服务器错误表示原始服务器出现问题。例如，HTTP 服务器繁忙或不可用。当出现服务器错误时，您的原始服务器将向 CloudFront 返回一个 500 区间内的 HTTP 状态代码，或者在某一时间段内 CloudFront 未从您的原始服务器获得响应，此时显示 504 状态代码（网关超时）。

CloudFront 可以为其返回自定义错误页面的 HTTP 状态代码包含以下各项：

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

有关 CloudFront 如何处理来自您的源的错误响应的详细说明，请参阅 CloudFront 如何处理与缓存 HTTP 4xx 和 5xx 状态码 (p. 88)。

创建或更新自定义错误页面的缓存行为

如果您希望将您的对象和自定义错误页面存储在不同的位置，您的分配必须包含满足以下条件时的缓存行为：

- Path Pattern (路径模式) 的值与您的自定义错误消息的路径匹配。例如，您在 Amazon S3 存储桶的 `/4xx-errors` 目录下为 4xx 错误保存了自定义错误页面。您的分配必须包含其路径模式将对您的自定义错误页面的请求路由至该位置的缓存行为，例如 /4xx-errors/*。
- Origin (源) 值指定包含您的自定义错误页面的源的 Origin ID (源 ID) 值。

有关更多信息，请参阅 您创建或更新 Web 分配时指定的值 (p. 29) 主题中的 缓存行为设置 (p. 33)。

更改响应代码

您可以选择 CloudFront 随自定义错误页面返回的 HTTP 状态代码作为给定的 HTTP 状态代码。例如，如果您的源返回 500 状态代码至 CloudFront，您可能希望 CloudFront 返回自定义错误页面和 200 状态代码 (OK) 至查看器。出于多种原因，您可能希望 CloudFront 返回的状态代码与您的源返回的状态代码不同：

- 一些 Internet 设备（例如，一些防火墙和企业代理）会拦截 HTTP 4xx 和 5xx 状态代码，防止响应返回到查看器。如果您替换 `200`，则通常不会拦截响应。
- 如果您不在意不同的客户端错误或服务器错误之间的区别，可以指定 400 或 500 作为 CloudFront 为所有 4xx 或 5xx 状态代码返回的值。
- 您可能希望返回 `200` 状态代码 (OK) 和静态网站，以便您的客户不知道您的网站宕机。

CloudFront 可以随自定义错误页面返回的 HTTP 状态代码包括以下各项：

- 200
- 400, 403, 404, 405, 414
- 500, 501, 502, 503, 504

控制 CloudFront 缓存错误的时长

默认情况下，当您的源返回 HTTP 4xx 或 5xx 状态代码时，CloudFront 将缓存这些错误响应五分钟，然后向您的源提交下一个对象请求以查看诱发错误的问题是否已得到解决以及请求的对象现在是否可用。

您可以为 CloudFront 缓存的每个 4xx 和 5xx 状态代码指定 (p. 73)错误缓存持续时间 —— Error Caching Minimum TTL (最短错误缓存 TTL)。指定持续时间时，注意以下几点：

- 如果您指定一个很短的错误缓存持续时间，则与您指定较长的持续时间相比，CloudFront 将向您的源转发更多请求。对于 5xx 错误，这可能会加重原先导致您的源返回错误的问题。
- 当您的源针对某个对象返回错误时，CloudFront 会通过错误响应或自定义错误页面响应对象请求，直到错误缓存持续时间过去。如果您指定一个很长的错误缓存持续时间，CloudFront 可能会在对象再次转为可用后的很长一段时间内继续使用错误响应或您的自定义错误页面来响应请求。

如果您要控制 CloudFront 为各个对象缓存错误的时长，可以配置您的原始服务器以为该对象的错误响应添加相应标题：

- 如果源添加 **`Cache-Control max-age`** 或 **`Cache-Control s-maxage`** 指令或 **`Expires`** 标题：则 CloudFront 缓存错误响应的时间将大于标题中的值或 Error Caching Minimum TTL (最短错误缓存 TTL) 的值。
- 如果源添加其他 **`Cache-Control`** 指令或不添加标题：则 CloudFront 缓存错误响应的时间等于 Error Caching Minimum TTL (最短错误缓存 TTL) 的值。

如果某一对象的 4xx 或 5xx 状态代码的过期时间超过您希望等待的时间，您可以使用所请求对象的 URL 使状态代码失效。如果您的源返回针对多个对象的错误响应，您需要分别使每个对象失效。有关使对象失效的更多信息，请参阅 使对象失效（仅 Web 分配） (p. 66)。

在自定义错误页面不可用时 CloudFront 如何响应

如果您配置 CloudFront 来为 HTTP 状态代码返回自定义错误页面，但自定义错误页面不可用，则 CloudFront 将向查看器返回 CloudFront 从包含自定义错误页面的源接收的状态代码。例如，假设您的自定义源返回 500 状态代码且您已将 CloudFront 配置为从 Amazon S3 存储桶获取 500 状态代码的自定义错误页面。

然而，某人意外将此自定义错误页面从您的存储桶中删除。在这种情况下，CloudFront 将返回 HTTP 状态代码 404（not found (未找到)）至请求对象的查看器。

自定义错误页面的定价

当 CloudFront 返回自定义错误页面至查看器时，您支付的是自定义错误页面的标准 CloudFront 费用，而不是所请求对象的费用。有关 CloudFront 费用的更多信息，请参阅 CloudFront 计费和使用 (p. 6)。

配置错误响应行为

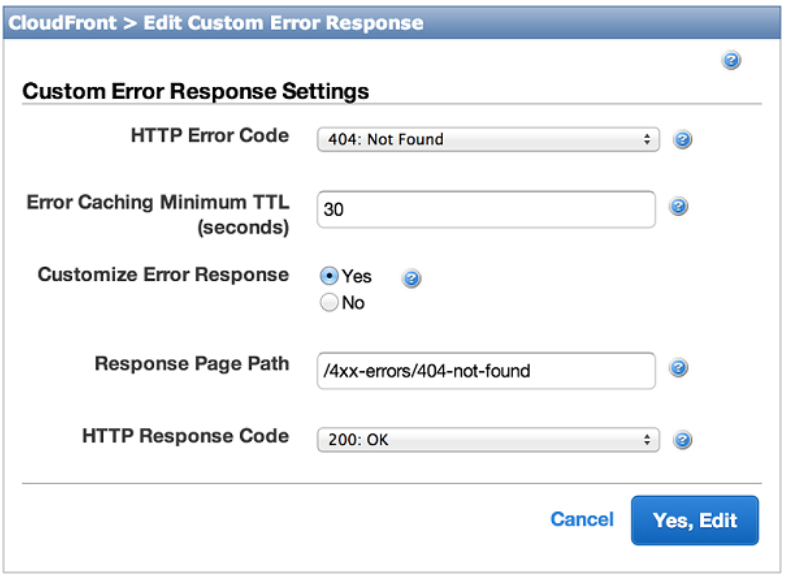
您可以使用 CloudFront API 或控制台来配置 CloudFront 错误响应。有关使用 CloudFront API 来配置错误响应的信息，请转到*Amazon CloudFront API 参考*中的 PUT 分配配置并参阅 `CustomErrorResponses` 元素。

要使用控制台配置 CloudFront 错误响应，请按以下步骤操作

1. 创建您希望 CloudFront 在您的源返回 HTTP 4xx 或 5xx 错误时向查看器返回的自定义错误页面。在可访问 CloudFront 的位置保存页面。

   我们建议您在 Amazon S3 存储桶中存储自定义错误页面，即使您使用自定义源也是如此。如果您在 HTTP 服务器上存储自定义错误页面，服务器启动以返回 5xx 错误，则由于原始服务器不可用，因此 CloudFront 将无法获取您希望返回至查看器的文件。
2. 确认您是否已至少向 CloudFront 授予对您的自定义错误页面的`read`权限。

   有关 Amazon S3 权限的更多信息，请参阅*Amazon Simple Storage Service 开发者指南*中的访问控制。有关使用 Amazon S3 控制台更新权限的信息，请转到*《Amazon Simple Storage Service 控制台用户指南》*。
3. （可选）配置您的原始服务器以随特定对象的错误响应添加`Cache-Control`指令或`Expires`标题，如适用。有关更多信息，请参阅 控制 CloudFront 缓存错误的时长 (p. 72)。
4. 登录 AWS 管理控制台，并通过以下网址打开 Amazon CloudFront 控制台：https://console.aws.amazon.com/cloudfront/。
5. 在分配列表中，选择要更新的分配并单击 Distribution Settings (分配设置)。
6. 单击 Error Pages (错误页面) 选项卡。然后单击 Create Custom Error Response (创建自定义错误响应)或选择现有错误代码并单击 Edit (编辑)。



7. 输入适用的值。有关更多信息，请参阅 自定义错误页面和错误缓存 (p. 39)。
8. 如果您将 CloudFront 配置为返回自定义错误页面，请添加或更新适当的缓存行为。有关更多信息，请参阅 创建或更新自定义错误页面的缓存行为 (p. 71)。
9. 要保存您所做的更改，请单击 Yes, Edit (是，编辑)。

CloudFront 如何处理对象的部分请求（范围 GET）

对于大型对象，最终用户的浏览器或客户端可能作出多个 GET 请求并使用 Range 请求标题以较小的单元下载对象。字节范围的这些请求，有时也被称为 Range GET 请求，提高了部分下载的效率并可恢复部分失败的传输。

当 CloudFront 接收 Range GET 请求时，它将检查接收请求的节点中的缓存。如果节点中的缓存已包含整个对象或所请求的部分对象，则 CloudFront 将立即从缓存提供所请求的范围。

如果缓存不包含所请求的范围，则 CloudFront 会将请求转发到源。（为优化性能，CloudFront 可请求比客户端在 Range GET 中请求的范围更大的范围。）接下来会发生什么取决于源是否支持 Range GET 请求：

- 如果源支持 **Range GET** 请求：它将返回请求的范围。CloudFront 提供请求的范围同时还会对其进行缓存以用于今后的请求。（Amazon S3 支持 Range GET 请求，就如一些 HTTP 服务器一样，例如，Apache 和 IIS。有关 HTTP 服务是否支持的信息，请参阅 HTTP 服务器的文档。）
- 如果源不支持 **Range GET** 请求：它将返回整个对象。CloudFront 提供整个对象并且还会缓存整个对象以用于未来请求。在 CloudFront 在边缘缓存中缓存整个对象后，它通过提供所请求的范围响应 Range GET 请求。

无论是哪种情况，CloudFront 都会在第一个字节到达后从源开始向最终用户提供所请求的范围或对象。

CloudFront 通常遵循 Range 标题的 RFC 规范。但是，如果您的 Range 标题不遵守以下要求，则 CloudFront 将返回 HTTP 状态代码 200 以及完整的对象，而不是状态代码 206 以及指定的范围：

- 范围必须列按升序列出。例如，`100-200,300-400` 是有效的，`300-400,100-200` 是无效的。
- 范围不能重叠。例如，`100-200,150-250` 是无效的。
- 所有范围的规范必须有效。例如，您不能指定负值作为范围的一部分。

有关 Range 请求标题的更多信息，请参阅*超文本传输协议 -- HTTP/1.1* 中的“第 14.35 节：范围”，网址为 http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35。

指定默认根对象（仅 Web 分配）

当最终用户请求分配的根 URL 而不是分配中的对象时，您可配置 CloudFront 以返回具体的对象（默认根对象）。指定一个默认根对象，以避免公开分配的内容。

例如，以下请求指向对象 `image.jpg`：

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

以下请求指向同一分配的根 URL 而不是具体的对象：

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

当您定义默认根对象时，调用分配的根的最终用户请求返回默认根对象。例如，如果您指定文件 `index.html` 作为您的默认根对象，请求：

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

返回：

```
http://d111111abcdef8.cloudfront.net/index.html
```

但是，如果您定义默认根对象，最终用户对分配的子目录的请求不返回默认根对象。例如，假设 `index.html` 是您的默认根对象且 CloudFront 接收最终用户对 CloudFront 分配下的 `install` 目录的请求：

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

CloudFront 将不会返回默认根对象，即使 `index.html` 的副本出现在 `install` 目录中也是如此。

如果您将自己的分配配置为允许 CloudFront 支持的所有 HTTP 方法，则默认根对象适用于所有方法。例如，如果您的默认根对象为 index.php 并且您编写应用程序以将 POST 请求提交至您的根域 (http://example.com)，则 CloudFront 会将请求发送到 http://example.com/index.php。

CloudFront 默认根对象的行为与 Amazon S3 索引文档的行为不同。当您配置 Amazon S3 存储桶作为网站并指定索引文档时，Amazon S3 将返回索引文档，即使用户请求存储桶中的子目录也是如此。（索引文档副本必须出现在每个子目录中。）有关配置 Amazon S3 存储桶作为网站以及有关索引文档的更多信息，请参阅*Amazon Simple Storage Service 开发者指南*中的在 Amazon S3 上托管网站一章。

Important

请记住，默认根对象仅适用于 CloudFront 分配。您仍需要管理源的安全。例如，如果您使用 Amazon S3 源，您仍需要适当地设置您的 Amazon S3 存储桶 ACL，以确保您想要的存储桶访问级别。

如果您不定义默认根对象，则对分配的根请求将传递到原始服务器。如果您使用 Amazon S3 源，则可能返回以下任何内容：

- 您的 Amazon S3 存储桶的内容列表— 在以下任何条件下，您的源的内容对使用 CloudFront 访问您分配的任何人可见：
  - 您的存储桶未正确配置。
  - 与您分配关联的存储桶和存储桶中对象的 Amazon S3 权限授予*每个人*访问权。
  - 最终用户使用源的根 URL 访问您的源。
- 您的源的私有内容列表— 如果您配置您的源作为私有分配（仅您和 CloudFront 有访问权），具有证书通过 CloudFront 访问您的分配的任何人可看见与您的分配关联的 Amazon S3 存储桶的内容。在这种情况下，用户不能通过源的根 URL 访问您的内容。有关分配私有内容的更多信息，请参阅 通过 CloudFront 提供私有内容。 (p. 91)。
- 错误 403 Forbidden (禁止访问)- 如果与您分配关联的 Amazon S3 存储桶的权限或该存储桶中对象的权限拒绝访问 CloudFront 和每个人，则 CloudFront 将返回此错误。

要避免暴露 Web 分配的内容或返回错误，请执行以下步骤以为您的分配指定默认根对象。

要为您的分配指定默认根对象，请按以下步骤操作

1. 将默认根对象上传到您的分配指向的源。

   文件可为 CloudFront 支持的任何类型。对于文件名的约束列表，请参阅 DistributionConfig 复杂类型中的 `DefaultRootObject` 元素说明。

Note

如果默认根对象的文件名太长或包含无效字符，则 CloudFront 将返回错误 `HTTP 400 Bad Request - InvalidDefaultRootObject`。此外，CloudFront 缓存代码五分钟，然后将结果写入访问日志。

2. 确认对象的权限至少授予 CloudFront `read` 访问权。

   有关 Amazon S3 权限的更多信息，请参阅*Amazon Simple Storage Service 开发者指南*中的访问控制。有关使用 Amazon S3 控制台更新权限的信息，请转到*《Amazon Simple Storage Service 控制台用户指南》*。
3. 更新您的分配，以使用 CloudFront 控制台或 CloudFront API 关连默认根对象。

   要使用 CloudFront 控制台指定默认根对象，请按以下步骤操作：

   a. 登录 AWS 管理控制台，并通过以下网址打开 Amazon CloudFront 控制台：https://console.aws.amazon.com/cloudfront/。
   b. 在顶部窗格中的分配列表中，选择要更新的分配。
   c. 在 Distribution Details (分配详细信息) 窗格中的 General (常规) 选项卡上，单击 Edit (编辑)。
   d. 在 Edit Distribution (编辑分配) 对话框的 Default Root Object (默认根对象) 字段中，输入默认根对象的文件名。

      仅输入对象名称，例如，`index.html`。请不要在对象名称前添加 `/`。
   e. 要保存您所做的更改，请单击 Yes, Edit (是，编辑)。

   要使用 CloudFront API 更新您的配置，您需要指定您分配中 `DefaultRootObject` 元素的值。有关使用 CloudFront API 指定默认根对象的信息，请参阅*Amazon CloudFront API 参考*中的 PUT 分配配置。
4. 确认您已经通过请求根 URL 启用了默认根对象。如果您的浏览器不显示默认根对象，请执行以下步骤：

   a. 通过在 CloudFront 控制台中查看分配的状态，确认您的分配已经完全部署。
   b. 重复第 2 步和第 3 步，验证您授予了正确的权限以及您正确地更新了分配的配置，以指定默认根对象。

提供压缩文件

Amazon CloudFront 可以从原始服务器提供压缩和未压缩的文件。CloudFront 依赖原始服务器来压缩文件或提供文件的压缩和未压缩的版本；CloudFront 不代表原始服务器执行压缩。凭借某些资格，CloudFront 还可以从 Amazon S3 提供压缩内容。有关更多信息，请参阅 选择要压缩的文件类型 (p. 79)。

提供压缩内容使下载更快，因为文件更小 — 在某些情况下，不到原件的一半大小。特别是对于 JavaScript 和 CSS 文件，更快的下载转化为向用户更快地提供网页。此外，由于 CloudFront 数据传输的费用基于提供的数据总量，因此提供压缩文件比提供未压缩文件更便宜。

如果查看器（例如，Web 浏览器或媒体播放器）通过将 `Accept-Encoding: gzip` 包含在请求标题中请求压缩内容，则 CloudFront 可以只提供压缩数据。必须使用 gzip 压缩内容，不支持其他压缩算法。如果请求标题包含额外的内容编码，例如，`deflate` 或 `sdch`，则 CloudFront 将在转发请求至原始服务器之前删除它们。如果 `Accept-Encoding` 字段中缺少 `gzip`，则 CloudFront 将仅提供文件的未压缩版本。有关 `Accept-Encoding` 请求标题字段的更多信息，请参阅*超文本传输协议 -- HTTP/1.1* 中“第 14.3 节：接受编码”，网址为 http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html。

有关更多信息，请参阅以下主题：

Topics

CloudFront 如何从自定义源提供压缩内容

以下介绍 CloudFront 通常如何从自定义源向 Web 应用程序提供压缩内容：

1. 您可以配置您的 Web 服务器以压缩选定的文件类型。有关更多信息，请参阅 选择要压缩的文件类型 (p. 79)。
2. 您可创建 CloudFront 分配。
3. 您可对您的 Web 应用程序进行编程，以使用 CloudFront URL 访问文件。
4. 用户在 Web 浏览器中访问您的应用程序。
5. CloudFront 将 Web 请求定向到对于用户来说具有最低延迟的节点，这在地理上可能是或可能不是最近的节点。
6. 在节点上，CloudFront 为每个请求中引用的对象检查缓存。如果浏览器在请求标题中包含 `Accept-Encoding: gzip`，则 CloudFront 将检查文件的压缩版本。如果未包含，则 CloudFront 将检查未压缩版本。
7. 如果文件在缓存中，则 CloudFront 将文件返回给 Web 浏览器。如果文件不在缓存中：
   a. 则 CloudFront 将请求转发到原始服务器。
   b. 如果请求针对您希望提供压缩的文件类型（参阅第 1 步），则 Web 服务器将压缩文件。
   c. Web 服务器将文件（压缩或未压缩，如适用）返回给 CloudFront。
   d. CloudFront 将文件添加到缓存并将文件提供给用户的浏览器。

当原始服务器运行 IIS 时提供压缩文件

默认情况下，IIS 不为经由代理服务器（如 CloudFront）的请求提供压缩内容。如果您使用 IIS 且将 IIS 配置为通过使用 `httpCompression` 元素压缩内容，请将 `noCompressionForHttp10` 和 `noCompressionForProxies` 属性的值更改为“false”。例如，对于 IIS 版本 7 及更高版本，您可以使用以下命令：

Important

在您运行这些命令之前，请参阅针对 `appcmd.exe` 的 IIS 文档以验证对于您的 IIS 版本，语法是否尚未发生更改。

```
%systemroot%\system32\inetsrv\appcmd.exe set config -section:system.webServer/ht
tpCompression /noCompressionForHttp10:false /commit:apphost
%systemroot%\system32\inetsrv\appcmd.exe set config -section:system.webServer/ht
tpCompression /noCompressionForProxies:false /commit:apphost
```

此外，如果您已经压缩对象，且该对象被请求的频率低于每隔几秒钟，您可能需要更改 `frequentHitThreshold` 和 `frequentHitTimePeriod` 的值。

有关更多信息，请参考 Microsoft 网站上的 IIS 文档。

当原始服务器运行 NGINX 时提供压缩文件

NGINX 的一些版本要求，您在使用 CloudFront 提供压缩文件时定制 NGINX 设置。在您所拥有的 NGINX 版本的文档中，如需有关以下设置的更多信息，请参阅文档的 `HttpGzipModule` 部分：

- gzip_http_version：CloudFront 以 HTTP 1.0 格式发送请求。在 NGINX 的某些版本中，`gzip_http_version` 设置的默认值为 1.1。如果您的 NGINX 版本包括此设置，请将其值更改为 `1.0`。
- gzip_proxied：当 CloudFront 将请求转发到原始服务器时，它包括 `Via` 标题。这导致 NGINX 根据代理方式解释请求，并且，默认情况下，NGINX 禁用代理请求的压缩。如果您的 NGINX 版本包括 `gzip_proxied` 设置，则将值更改为 `any`。

从 Amazon S3 提供压缩文件

如果您想从 Amazon S3 提供压缩文件：

1. 为每个文件创建两个版本，一个压缩版本，一个未压缩版本。要确保压缩和未压缩文件版本在 CloudFront 缓存中不相互覆盖，请为每个文件指定一个唯一的名称，例如，welcome.js 和 welcome.js.gz。
2. 通过以下网址打开 Amazon S3 控制台：https://console.aws.amazon.com/s3/。
3. 将两个版本都上传到 Amazon S3。
4. 为每个压缩文件添加 `Content-Encoding` 标题字段，并将该字段的值设置为 `gzip`。

   有关如何使用适用于 PHP 的 AWS 开发工具包添加 `Content-Encoding` 标题字段的示例，请参阅 *Amazon Simple Storage Service 开发者指南*中的使用适用于 PHP 的 AWS 开发工具包上传对象。一些第三方工具也可用于添加此字段。

   要添加 `Content-Encoding` 标题字段并使用 Amazon S3 控制台设置字段值，请执行以下步骤：

   a. 在 Amazon S3 控制台中，在“存储桶”窗格里，单击包含压缩文件的存储桶的名称。
   b. 在“Objects and Folders (对象和文件夹)”窗格的顶部，单击 Actions (操作)，然后在 Actions (操作) 列表中，单击 Properties (属性)。
   c. 在“Properties (属性)”窗格中，单击 Metadata (元数据) 选项卡。
   d. 在“Objects and Folders (对象和文件夹)”窗格中，单击您想为其添加 `Content-Encoding` 标题字段的文件的名称。
   e. 在“Metadata (元数据)”选项卡上，单击 `Add More Metadata` (添加更多元数据)。
   f. 在“Key (键)”列表中，单击 `Content-Encoding`。
   g. 在“Value (值)”字段中，输入 `gzip`。
   h. 单击 Save (保存)。
   i. 为剩余的压缩文件重复第 4d 步到第 4h 步。

5. 当生成链接到 CloudFront 中内容的 HTML 时（例如，使用 php、asp 或 jsp），评估来自查看器的请求是否在请求标题中包括 `Accept-Encoding: gzip`。如果包括，请改写相应的链接，以指向压缩的对象名称。

选择要压缩的文件类型

有些类型的文件压缩效果很好，例如，HTML、CSS 和 JavaScript 文件。有些类型的文件可压缩百分之几，但还不足以证明您需要额外的处理器周期来让您的 Web 服务器压缩内容，且有些类型的文件在压缩后甚至会变得更大。通常压缩效果不好的文件类型包括已压缩的图像文件（.jpg、.gif），视频格式以及音频格式。我们建议您在您的分配中测试文件类型的压缩，以确保压缩有足够的优势。

请求和响应行为

下面各节介绍了 CloudFront 如何处理最终用户的请求及如何将请求转发给您的 Amazon S3 或自定义源，以及 CloudFront 如何处理来自源的响应，包括 CloudFront 如何处理与缓存 4xx 和 5xx HTTP 状态码。

Topics

Amazon S3 源的请求和响应行为

Topics

CloudFront 如何处理请求及如何将请求转发给您的 Amazon S3 原始服务器

有关 CloudFront 如何处理最终用户的请求及如何将请求转发给您的 Amazon S3 源的信息，请参阅相关主题：

Topics

缓存持续时间和最小 TTL

对于 Web 分配，要控制对象在 CloudFront 缓存中保留多长时间后 CloudFront 便会将另一请求转发到您的源，您可以：

- 将您的源配置为向每个对象添加一个 `Cache-Control` 或 `Expires` 标头字段。
- 指定 CloudFront 缓存行为的最小 TTL 值。
- 使用默认值 24 小时。

有关更多信息，请参阅 指定对象在 CloudFront 边缘缓存中的保留时间（过期）(p. 63)。

有条件 GET

当 CloudFront 从节点缓存收到针对已过期对象的请求时，它会将该请求转发到 Amazon S3 源，以获取相应对象的最新版本，或从 Amazon S3 确认 CloudFront 节点缓存已包含最新版本。当 Amazon S3 最初向 CloudFront 发送对象时，它会在响应中包含一个 `ETag` 值和一个 `LastModified` 值。在 CloudFront 转发给 Amazon S3 的新请求中，CloudFront 会添加以下两项或其中一项：

- `If-Match` 或 `If-None-Match` 标头，其中包含该对象已过期版本的 `ETag` 值。
- `If-Modified-Since` 标头，其中包含该对象已过期版本的 `LastModified` 值。

Amazon S3 使用此信息来确定该对象是否已更新，并据此确定是将整个对象返回给 CloudFront 还是只返回 HTTP 304 状态码（未修改）。

Cookie

Amazon S3 不处理 Cookie。如果您将缓存行为配置为将 Cookie 转发到 Amazon S3 源，则 CloudFront 将转发 Cookie，而 Amazon S3 则会忽略它们。

跨源资源共享 (CORS)

CloudFront 目前不支持 Amazon S3 跨源资源共享。

HTTP 方法

如果您将 CloudFront 配置为处理其支持的所有 HTTP 方法，则 CloudFront 会接受来自用户的以下请求，并将这些请求转发给您的 Amazon S3 源：

- `DELETE`
- `GET`
- `HEAD`
- `OPTIONS`
- `PATCH`
- `POST`
- `PUT`

CloudFront 会缓存对 `GET` 和 `HEAD` 请求的响应，而不会缓存对使用其他方法的请求的响应。

如果您想使用多部分上传方式来将对象添加到 Amazon S3 存储桶，则必须将 CloudFront 原始访问标识添加到您的分配中，并向原始访问标识授予适用的许可。有关更多信息，请参阅 使用原始访问标识限制访问您的 Amazon S3 内容 (p. 95)。

Caution

如果您将 CloudFront 配置为接受 CloudFront 支持的所有 HTTP 方法并将这些方法转发到 Amazon S3，则必须创建一个 CloudFront 原始访问标识，以限制对您 Amazon S3 内容的访问，并向该原始访问标识授予适用的许可。例如，如果因为您想使用 PUT 而将 CloudFront 配置为接受并转发这些方法，则必须将 Amazon S3 存储桶策略或 ACL 配置为适当处理 DELETE 请求，以便用户无法删除您不希望其删除的资源。有关更多信息，请参阅 使用原始访问标识限制访问您的 Amazon S3 内容 (p. 95)。

有关 Amazon S3 所支持的操作的信息，请参阅 Amazon S3 文档。

IP 地址

CloudFront 转发给 Amazon S3 的 IP 地址是 CloudFront 服务器的 IP 地址，而非最终用户的计算机的 IP 地址。

请求的最大长度与 URL 的最大长度

请求的最大长度（包括路径、查询字符串（如果有）以及标头）为 20480 个字节。

CloudFront 根据请求来构造 URL。此 URL 的最大长度是 8192 个字节。

如果请求或 URL 超出了这两个上限，则 CloudFront 将丢弃这样的请求。

协议

CloudFront 根据最终用户发送到 CloudFront 的请求的协议（HTTP 或 HTTPS），来向原始服务器转发 HTTP 或 HTTPS 请求。

查询字符串

对于 Web 分配，您可配置是否让 CloudFront 将查询字符串参数转发到您的 Amazon S3 源。对于 RTMP 分配，CloudFront 不转发查询字符串参数。有关更多信息，请参阅 CloudFront 如何转发、缓存及记录查询字符串参数 (p. 59)。

CloudFront 如何处理来自 Amazon S3 原始服务器的响应

Topics

已取消的请求

如果对象不在节点缓存中，或者在 CloudFront 从您的源获取对象后，还未来得及传送请求的对象，查看器便终止了会话（例如关闭浏览器），则 CloudFront 不会将该对象缓存在节点中。

跨源资源共享 (CORS)

CloudFront 目前不支持 Amazon S3 跨源资源共享。

最大文件大小

CloudFront 将返回给最终用户的响应正文的最大大小为 20 GB。这包括未指定 `Content-Length` 标头值的分块传输响应。

重定向

您可将 Amazon S3 存储桶配置为将所有请求重定向到另一个主机名；该主机名可以是另一个 Amazon S3 存储桶，也可是一台 HTTP 服务器。如果您将存储桶配置为重定向所有请求，并且该存储桶是用于 CloudFront 分配的源，我们建议您将该存储桶配置为将所有请求重定向到 CloudFront 分配时，使用该分配的域名（例如，d111111abcdef8.cloudfront.net）或与该分配关联的备用域名（别名记录，例如 example.com）。否则，最终用户请求会绕过 CloudFront，直接从新源提供对象。

Note

如果您将请求重定向到备用域名，您还必须通过添加别名记录为您的域更新 DNS 服务。有关更多信息，请参阅 使用备用域名（别名记录）(p. 52)。

当您将存储桶配置为重定向所有请求时，将发生以下情况：

1. 查看器（例如，浏览器）向 CloudFront 请求对象。
2. CloudFront 将请求转发到作为分配源的 Amazon S3 存储桶。
3. Amazon S3 返回 HTTP 状态码 301（已永久移动）以及新位置。
4. CloudFront 缓存重定向状态码和新位置，并将值返回给查看器。CloudFront 不会按照重定向路线从新位置获取对象。
5. 查看器再发送一个针对该对象的请求，但查看器此次会指定其从 CloudFront 获取的新位置：
   - 如果 Amazon S3 存储桶将所有请求重定向到 CloudFront 分配时使用的是分配的域名或备用域名，则 CloudFront 会向位于新位置的 Amazon S3 存储桶或 HTTP 服务器请求对象。当新位置返回对象时，CloudFront 将其返回给查看器并将其缓存在节点中。
   - 如果 Amazon S3 存储桶是将请求重定向到另一个位置，则第二个请求会绕过 CloudFront。位于新位置的 Amazon S3 存储桶或 HTTP 服务器会将该对象直接返回给查看器，因此，该对象绝不会缓存在 CloudFront 节点缓存中。

自定义源的请求和响应行为

Topics

CloudFront 如何处理请求及如何将请求转发给您的自定义原始服务器

有关 CloudFront 如何处理最终用户的请求及如何将请求转发给自定义源的信息，请参阅相关主题：

Topics

身份验证

切勿将您的原始服务器配置为请求客户端身份验证，因为 CloudFront 无法将证书从查看器转发到您的源。您可以将 CloudFront 配置为使用 HTTP 或 HTTPS 将请求转发到源；有关更多信息，请参阅如何要求在查看器、CloudFront 和您的源之间采用 HTTPS 进行通信 (p. 135)。

缓存持续时间和最小 TTL

对于 Web 分配，要控制对象在 CloudFront 缓存中保留多长时间后 CloudFront 便会将另一请求转发到您的源，您可以：

- 将您的源配置为向每个对象添加一个 `Cache-Control` 或 `Expires` 标头字段。
- 指定 CloudFront 缓存行为的最小 TTL 值。
- 使用默认值 24 小时。

有关更多信息，请参阅 指定对象在 CloudFront 边缘缓存中的保留时间（过期）(p. 63)。

压缩

CloudFront 会转发包含 `Accept-Encoding` 字段值 `"identity"` 和 `"gzip"` 的请求。有关更多信息，请参阅 提供压缩文件 (p. 76)。

有条件请求

当 CloudFront 从节点缓存收到针对已过期对象的请求时，它会将该请求转发到源，以获取该对象的最新版本，或从源确认 CloudFront 节点缓存已包含最新版本。通常，当源最后一次将对象发送到 CloudFront 时，它会在响应中包含一个 `ETag` 值和/或一个 `LastModified` 值。在 CloudFront 转发给源的新请求中，CloudFront 会添加以下两项或其中一项：

- `If-Match` 或 `If-None-Match` 标头，其中包含该对象已过期版本的 `ETag` 值。
- `If-Modified-Since` 标头，其中包含该对象已过期版本的 `LastModified` 值。

源使用此信息来确定该对象是否已更新，并据此确定是将整个对象返回给 CloudFront 还是仅返回 HTTP 304 状态码（未修改）。

Cookie

您可将 CloudFront 配置为将 Cookie 转发到您的源。有关更多信息，请参阅 CloudFront 如何转发、缓存及记录 Cookie (p. 61)。

加密

CloudFront 使用 SSLv3 或 TLSv1 协议以及 AES128-SHA1 或 RC4-MD5 密码将 HTTPS 请求转发给原始服务器。如果您的原始服务器不支持 AES128-SHA1 或 RC4-MD5 密码，则 CloudFront 无法建立到该源的 SSL 连接。

建立到源的 HTTPS 连接时，CloudFront 会添加一个服务器名称指示 (SNI) 扩展标记，并包含您的分配的适用源域名的值。有关 SNI 的更多信息，请参阅 RFC 4366，传输层安全性 (TLS) 扩展标记的第 3.1 节。

HTTP 方法

如果您将 CloudFront 配置为处理其支持的所有 HTTP 方法，则 CloudFront 会接受来自用户的以下请求，并将这些请求转发给您的自定义源：

- `DELETE`
- `GET`
- `HEAD`
- `OPTIONS`
- `PATCH`
- `POST`
- `PUT`

CloudFront 会缓存对 `GET` 和 `HEAD` 请求的响应，而不会缓存对使用其他方法的请求的响应。

有关如何配置是否让您的自定义源处理这些方法的信息，请参阅该源的文档。

Caution

如果您将 CloudFront 配置为接受 CloudFront 支持的所有 HTTP 方法并将这些方法转发到您的源，请将您的原始服务器配置为处理所有方法。例如，如果因为您想使用 `POST` 而将 CloudFront 配置为接受并转发这些方法，则必须将您的原始服务器配置为适当处理 `DELETE` 请求，以便用户无法删除您不希望其删除的资源。有关更多信息，请参阅您的 HTTP 服务器的文档。

HTTP 版本

CloudFront 使用 HTTP/1.0 将请求转发到您的自定义源，但支持大多数 HTTP 1.1 规范。为提升性能，我们建议您将 `Keep-Alive` 标头包含在最终用户的请求中。

IP 地址

CloudFront 转发到原始服务器的 IP 地址是 CloudFront 服务器的 IP 地址，而不是最终用户计算机的 IP 地址。

请求的最大长度与 URL 的最大长度

请求的最大长度（包括路径、查询字符串（如果有）以及标头）为 20480 个字节。

CloudFront 根据请求来构造 URL。此 URL 的最大长度是 8192 个字节。

如果请求或 URL 超出了这两个上限，则 CloudFront 将丢弃这样的请求。

协议

CloudFront 根据以下内容将 HTTP 或 HTTPS 请求转发到原始服务器：

- 最终用户发送到 CloudFront 的请求的协议（HTTP 或 HTTPS）。
- CloudFront 控制台中 Origin Protocol Policy (源协议策略) 字段的值，或（如果您使用的是 CloudFront API）`DistributionConfig` 复合类型中的 `OriginProtocolPolicy` 元素。在 CloudFront 控制台中，对应的选项为 HTTP Only (仅限 HTTP) 和 Match Viewer (匹配查看器)。

如果您指定 HTTP Only (仅限 HTTP)，则 CloudFront 仅使用 HTTP 协议将请求转发到原始服务器，而不考虑最终用户请求中的协议。

如果您指定 Match Viewer (匹配查看器)，则 CloudFront 将使用最终用户请求中的协议将请求转发到原始服务器。请注意，CloudFront 仅缓存对象一次，即使查看器同时使用 HTTP 和 HTTPS 协议发出请求也是如此。

Caution

如果最终用户请求使用 HTTPS 协议，或者原始服务器返回无效证书或自签名证书，CloudFront 将中断 TCP 连接。

如果不确定使用哪种协议，我们建议您指定“HTTP only (仅限 HTTP)”。

有关如何使用 CloudFront 控制台更新分配的信息，请参阅列出、查看及更新 CloudFront 分配 (p. 55)。有关如何使用 CloudFront API 更新分配的信息，请访问 *Amazon CloudFront API 参考*中的 PUT 分配配置。

查询字符串

您可配置是否让 CloudFront 将查询字符串参数转发到您的源。有关更多信息，请参阅CloudFront 如何转发、缓存及记录查询字符串参数 (p. 59)。

已删除的标头字段

在将请求转发到您的源之前，CloudFront 将删除逐跳标头字段，如 `Authorization` 和 `Connection` 字段。

User-Agent 标头

CloudFront 会先在 `User-Agent` 标头中添加以下值，再将请求转发到您的源：

```
User-Agent = Amazon CloudFront
```

无论来自查看器的请求是否包含了 `User-Agent` 标头，CloudFront 均添加此标头。如果来自查看器的请求包含 `User-Agent` 标头，则 CloudFront 会删除它。

CloudFront 如何处理来自自定义原始服务器的响应

有关 CloudFront 如何处理来自自定义原始服务器的响应的信息，请参阅相关主题：

Topics

缓存

- 确保原始服务器为 `Date` 和 `Last-Modified` 标头字段设置了有效、准确的值。
- 如果来自最终用户的请求包含 `If-Match` 或 `If-None-Match` 请求标头字段，请设置 `ETag` 响应标头字段。如果您没有指定 `ETag` 值，则 CloudFront 将忽略后续的 `If-Match` 或 `If-None-Match` 标头。

已取消的请求

如果对象不在节点缓存中，或者在 CloudFront 从您的源获取对象后，还未来得及传送请求的对象，查看器便终止了会话（例如关闭浏览器），则 CloudFront 不会将该对象缓存在节点中。

内容协商

`Vary` 标头唯一接受的值为 `Accept-Encoding`。CloudFront 会忽略其他值。

Cookie

如果您为缓存行为启用了 Cookie，或者源返回包含对象的 Cookie，则 CloudFront 会将该对象和 Cookie 都缓存下来。请注意，这会降低对象的可缓存性。有关更多信息，请参阅 CloudFront 如何转发、缓存及记录 Cookie (p. 61)。

中断的 TCP 连接

如果在您的源将对象返回给 CloudFront 时，CloudFront 和您的源之间的 TCP 连接中断，那么 CloudFront 的行为将取决于您的源是否在响应中包含 `Content-Length` 标头：

- 有 Content-Length 标头：CloudFront 在从您的源获得对象后会将对象原样返回给查看器。但是，如果 `Content-Length` 标头的值与对象大小不匹配，CloudFront 则不缓存对象。
- 无 Content-Length 标头：CloudFront 将对象返回给查看器并对其进行缓存，但该对象可能不完整。在无 `Content-Length` 标头的情况下，CloudFront 无法确定 TCP 连接是意外中断的还是有意中断的。

我们建议您将 HTTP 服务器配置为添加一个 `Content-Length` 标头来防止 CloudFront 缓存部分对象。

最大文件大小

CloudFront 将返回给最终用户的响应正文的最大大小为 20 GB。这包括未指定 `Content-Length` 标头值的分块传输响应。

源不可用

如果您的原始服务器不可用且 CloudFront 获得针对节点缓存中已过期对象的请求（例如，因为 `Cache-Control max-age` 指令中指定的期限已过），那么 CloudFront 会提供该对象的已过期版本或提供自定义错误页面。有关更多信息，请参阅 CloudFront 如何处理与缓存 HTTP 4xx 和 5xx 状态码 (p. 88)。

某些情况下，会逐出很少被请求的对象，不再将它们存放在节点缓存中。CloudFront 无法提供已被逐出的对象。

重定向

如果您更改对象在原始服务器上的位置，则您可将 Web 服务器配置为将请求重定向到新位置。配置重定向后，最终用户第一次提交针对对象的请求时，CloudFront 会将请求发送到源，源通过重定向（例如，`302 Moved Temporarily`）加以响应。CloudFront 会缓存重定向路线并将其返回给最终用户。CloudFront 不会按照重定向路线进行跟踪。

您可以将您的 Web 服务器配置为将请求重定向到以下位置之一：

- 对象在原始服务器上的新 URL。当最终用户按照重定向路线访问新 URL 时，最终用户将绕过 CloudFront，直接到达源。因此，我们建议您不要将请求重定向到对象在源上的新 URL。
- 对象的新 CloudFront URL。当最终用户提交包含新 CloudFront URL 的请求时，CloudFront 将从源上的新位置获取对象，并将其缓存在节点中，然后将该对象返回给最终用户。针对该对象的后续请求将由节点提供。这避免了与查看器向源请求对象相关的延迟和负载。但是，针对该对象的每个新请求都将因向 CloudFront 发送两个请求而产生费用。

传输编码

CloudFront 仅支持 `Transfer-Encoding` 标头的 `chunked` 值。如果您的源返回 `Transfer-Encoding: chunked`，则 CloudFront 会在节点收到区块时把区块组合成一个整体对象，并将该对象返回给客户端，然后将该对象缓存起来以提供给后续请求。此外，CloudFront 将计算 `Content-Length` 标头并在响应后续的客户端请求时返回该标头。

我们建议您不要使用分块传输编码。如果区块发生延迟（例如，因防火墙原因），CloudFront 将无从得知其是否已经获得对象中的最后一个区块，并且可能关闭连接而仅缓存部分对象。有关更多信息，请参阅中断的 TCP 连接 (p. 87)。

CloudFront 如何处理与缓存 HTTP 4xx 和 5xx 状态码

Topics

当 CloudFront 向您的 Amazon S3 存储桶或自定义原始服务器请求对象时，您的源有时会返回 HTTP 4xx 或 5xx 状态码，这表示已发生错误。CloudFront 的行为取决于：

- 您是否已配置自定义错误页面。
- 您是否已配置希望 CloudFront 缓存来自源的错误响应的时长（最短错误缓存 TTL）。
- 状态码。
- 对于 5xx 状态码，请求的对象当前是否在 CloudFront 节点缓存中。

有关在 CloudFront 控制台中设置自定义错误页面的信息，请参阅自定义错误页面和错误缓存 (p. 39)。有关 CloudFront 控制台中最短错误缓存 TTL 的信息，请参阅错误缓存最小 TTL (p. 39)。

有关 CloudFront 缓存的 HTTP 状态码的列表，请参阅 CloudFront 缓存的 HTTP 4xx 和 5xx 状态码 (p. 89)。

如果您已启用日志记录，无论 HTTP 状态码是什么，CloudFront 均会将结果写入日志中。

当您已配置自定义错误页面时 CloudFront 如何处理错误

当您的源返回 HTTP 4xx 或 5xx 状态码且您已配置自定义错误页面时，CloudFront 会执行以下操作：

1. 在已收到源请求的 CloudFront 节点缓存中，CloudFront 会检查您的分配配置，并获取与原始服务器返回的状态码对应的自定义错误页面的路径。
2. CloudFront 在您的分配中查找路径模式与自定义错误页面的路径相匹配的第一项缓存行为。
3. CloudFront 节点会将针对自定义错误页面的请求发送到在缓存行为中指定的源。
4. 源将自定义错误页面返回给节点。
5. CloudFront 将自定义错误页面返回给发出请求的查看器，还会将自定义错误页面缓存最短错误缓存 TTL 所指定的时长（默认为五分钟）。
6. 在最短错误缓存 TTL 过后，CloudFront 会通过再向您的源转发一个请求来再次尝试获取请求的对象。

当您尚未配置自定义错误页面时 CloudFront 如何处理错误

当请求的对象*未*在节点缓存中时，如果您的源返回 HTTP 4xx 状态码或返回 HTTP 5xx 状态码，并且您尚未配置自定义错误页面，则 CloudFront 会执行以下操作：

1. CloudFront 会将 4xx 或 5xx 状态码返回给查看器。
2. CloudFront 也会在收到请求的节点缓存中缓存状态码。
3. 在最短错误缓存 TTL 的持续时间内（默认为五分钟），CloudFront 会以缓存的 4xx 或 5xx 状态码来响应针对同一对象的后续查看器请求。
4. 在最短错误缓存 TTL 过后，CloudFront 会通过再向您的源转发一个请求来再次尝试获取请求的对象。

当您的源返回 HTTP 5xx 状态码，请求的对象*在*节点缓存中，且您尚未配置自定义错误页面时，CloudFront 会执行以下操作：

1. 如果请求的对象仍在节点缓存中，那么即使它已过期，CloudFront 也会提供该对象。
2. 在最短错误缓存 TTL 的持续时间内，CloudFront 会通过从节点缓存提供对象来响应针对同一对象的后续查看器请求。
3. 在最短错误缓存 TTL 过后，CloudFront 会通过再向您的源转发一个请求来再次尝试获取请求的对象。请注意，如果未频繁请求该对象，当您的原始服务器仍返回 5xx 响应时，CloudFront 可能会将该对象从节点缓存中逐出。有关对象在 CloudFront 节点缓存中的保留时长的信息，请参阅指定对象在 CloudFront 边缘缓存中的保留时间（过期）(p. 63)。

CloudFront 缓存的 HTTP 4xx 和 5xx 状态码

CloudFront 会缓存由 Amazon S3 或您的自定义原始服务器返回的以下 HTTP 4xx 和 5xx 状态码。如果您已针对某一 HTTP 状态码配置自定义错误页面，则 CloudFront 会缓存该自定义错误页面。

400	错误请求
403	禁止
404	未找到

405	不允许的方法
414	请求 URI 太大
500	内部服务错误
501	未实现
502	无效网关
503	服务不可用
504	网关超时

通过 CloudFront 提供私有内容。

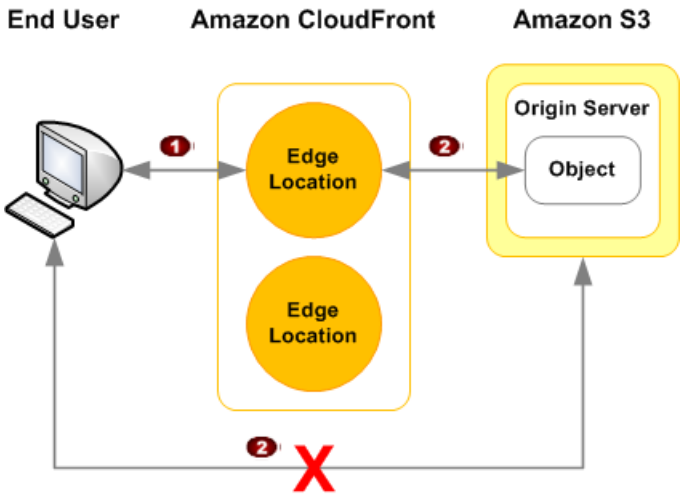
很多通过互联网分发内容的公司都希望限制对文档、业务数据、媒体流或仅供部分用户（例如已付费的用户）使用的内容的访问。要使用 CloudFront 安全地提供此类私有内容，您可以：

- 要求您的用户使用经过 CloudFront 签名的专用 URL 来访问您的内容，而不是使用标准的 CloudFront 公共 URL。
- 要求您的用户使用 CloudFront URL 而非 Amazon S3 URL 来访问您的 Amazon S3 内容。

私有内容概述

您可通过两种方式控制最终用户对私有内容的访问：

1	您可限制对 CloudFront 节点缓存中对象的访问：可以将 CloudFront 配置为要求最终用户使用专用的签名 URL 访问您的对象。然后，您就可以创建签名 URL（手动或以编程方式）并将他们分发给您的用户。 当您为对象创建签名 URL 时，您可指定： • 结束日期和时间，在此之后，URL 不再有效。 • （可选）URL 生效的日期和时间。 • （可选）可用于访问您内容的计算机的 IP 地址或地址范围。 签名 URL 中有一部分是使用公钥/私钥对中的私钥进行散列和签署的。当某人使用签名 URL 来访问对象时，CloudFront 将比较该 URL 中已签名和未签名的部分。如果两者不匹配，CloudFront 将不提供对象。
2	您可限制对您 Amazon S3 存储桶中对象的访问：可以对您 Amazon S3 存储桶中的内容进行保护，使最终用户可使用 CloudFront URL 访问它，但不能使用 Amazon S3 URL 加以访问。此举可防止任何人绕过 CloudFront 而使用 Amazon S3 URL 访问您试图限制访问的内容。如果要求用户使用 CloudFront URL，您需要： • 创建一个称作原始访问标识的专用 CloudFront 用户。 • 向该原始访问标识授予读取您存储桶中对象的权限。 • 删除任何其他人读取这些对象的权限。



私有内容的工作原理

下文概述了如何使用私有内容来保护对您 Amazon S3 内容的访问。稍后，我们将对每一步进行更详细的介绍。

Note

要设置私有内容，您必须使用 CloudFront 控制台或 CloudFront API 版本 2009-09-09 或更高版本。

1. 对 Amazon S3 中的内容进行保护，以防止任何人绕过 CloudFront 而使用 Amazon S3 URL 访问您试图限制访问的内容。此步骤是可选的，但最好完成这一步，以防有人获知您内容的 Amazon S3 URL。

   a. 创建一个原始访问标识，这是一种专用的 CloudFront 用户。
   b. 将原始访问标识与您的分配关联起来。（对于 Web 分配，您需要将原始访问标识与源关联起来，以便您可以保护您的所有 Amazon S3 内容或者只保护其中一部分内容。）
   c. 更改 Amazon S3 中的权限，以便只有原始访问标识可访问您的对象。

   有关更多信息，请参阅 使用原始访问标识限制访问您的 Amazon S3 内容 (p. 95)。
2. 在您的 CloudFront 分配中，指定一个或多个可信签署人，即您希望有权创建签名 URL 的 AWS 账户。

   有关更多信息，请参阅 指定可创建签名 URL 的 AWS 账户（可信签署人）(p. 99)。
3. 开发您的应用程序，以便为对象创建签名 URL，或者针对应用程序中需要签名 URL 的部分进行开发。

   有关签名 URL 的更多信息，请参阅签名 URL 概述 (p. 104)。
4. 最终用户请求您希望要求采用签名 URL 的对象。
5. 您的应用程序验证最终用户是否有权访问该对象：他们是否已注册、是否已支付了内容访问费用或者是否已满足一些其他的访问要求。
6. 您的应用程序创建签名 URL 并将其发回用户。
7. 通过使用签名 URL，用户可以下载内容或对内容进行流式处理。

此步骤自动完成；用户通常不必执行任何额外的操作即可访问内容。例如，如果用户是在 Web 浏览器中访问您的内容，那么您的应用程序会将签名 URL 发回浏览器。浏览器可以直接使用签名 URL 来访问 CloudFront 节点缓存中的对象，而无需用户进行任何干预。

8. CloudFront 确认该 URL 尚未被篡改，并且依然有效。例如，如果您为该 URL 指定了开始和结束日期及时间，CloudFront 会确认用户是否在您允许访问的时间段尝试访问您的内容。如果该 URL 有效，CloudFront 将执行标准操作：确定对象是否已在节点缓存中，必要时将请求转发到源，然后将对象发回用户。

对私有内容使用 HTTP 服务器

您可以对任何 CloudFront 分配使用签名 URL，而不考虑相应的源是 Amazon S3 存储桶还是 HTTP 服务器。但是，要使 CloudFront 能够访问 HTTP 服务器上的对象，这些对象必须保持可公开访问状态。由于这些对象可公开访问，因此如果没有由 CloudFront 签名 URL 提供的保护，任何人员只要拥有 HTTP 服务器上对象的 URL，就可以访问它们。如果您使用签名 URL 且您的源是 HTTP 服务器，切勿将您 HTTP 服务器上对象的 URL 提供给您的客户或您组织外的其他人。

选择签名 URL 的有效期限

您可以使用只有很短有效期（可能只有几分钟）的签名 URL 来分发私有内容。有效期如此短的签名 URL 适合动态地向终端用户分发内容供其用于有限的用途，例如按需向客户分发出租电影或音乐下载。如果您的签名 URL 的有效期很短，您将可能希望使用您开发的应用程序自动生成它们。当用户开始下载对象或开始播放媒体文件时，CloudFront 将比较 URL 中的过期时间和当前时间，以确定该 URL 是否仍然有效。

您也可使用有效期较长（可能数年）的签名 URL 来分发私有内容。有效期较长的签名 URL 适合将私有内容分配给已知的最终用户，例如向投资者分发业务计划书或向员工分发培训材料。您可以开发一款应用程序来为您生成这些有效期较长的签名 URL，您也可使用用于配置私有内容的工具 (p. 251)中所列的第三方 GUI 工具之一。

示例代码和第三方工具

有关创建签名 URL 散列和签署部分的示例代码，请参阅以下主题：

- 使用 Perl 创建 URL 签名 (p. 120)
- 使用 PHP 创建 URL 签名 (p. 121)
- 使用 C# 和 .NET Framework 创建 URL 签名 (p. 123)
- 使用 Java 创建 URL 签名 (p. 131)

Amazon CloudFront 示例代码和库页面上提供了更多有关创建签名 URL 的示例代码。

有关支持私有内容（包括创建签名 URL）的第三方工具的信息，请参阅用于配置私有内容的工具 (p. 251)。

任务列表：提供私有内容

要配置 CloudFront 以提供私有内容，请执行以下任务。

1. *可选：*将您的 CloudFront 分配和 Amazon S3 存储桶配置为要求您的用户仅使用 CloudFront URL 来访问您的 Amazon S3 内容。有关更多信息，请参阅 使用原始访问标识限制访问您的 Amazon S3 内容 (p. 95)。
2. 指定您希望用于创建签名 URL 的 AWS 账户。有关更多信息，请参阅 指定可创建签名 URL 的 AWS 账户（可信签署人） (p. 99)。
3. 手动或以编程方式创建要向授权的最终用户提供的签名 URL。有关更多信息，请参阅 签名 URL 概述 (p. 104)。

使用原始访问标识限制访问您的 Amazon S3 内容

Topics

通常，如果您使用 Amazon S3 存储桶作为 CloudFront 分配的源，那么您需向每个人授予读取您存储桶中对象的权限。这样，任何人都可使用 CloudFront URL 或 Amazon S3 URL 访问您的对象。CloudFront 不会公开 Amazon S3 URL，但如果您的应用程序服务器直接从 Amazon S3 提供任何对象，或者有任何人公布指向 Amazon S3 中特定对象的直接链接，那么您的用户就可能会获得这些 URL。

如果您希望使用 CloudFront 签名 URL 来提供对 Amazon S3 存储桶中对象的访问权，您可能还需要防止用户使用 Amazon S3 URL 访问您的 Amazon S3 对象。如果用户直接在 Amazon S3 中访问您的对象，他们会绕过 CloudFront 签名 URL 提供的控制机制，包括用来控制 URL 何时过期的机制以及用来控制可使用哪些 IP 地址来访问对象的机制。此外，如果用户使用 CloudFront URL 和 Amazon S3 URL 访问对象，CloudFront 访问日志的用处就不太大，因为它们是不完整的。

您需通过创建原始访问标识（是一种专用的 CloudFront 用户）来限制访问 Amazon S3 内容。可以通过更改 Amazon S3 权限来向原始访问标识提供访问您对象的权限，并删除所有其他人的权限。当您的用户使用 CloudFront URL 访问您的 Amazon S3 对象时，CloudFront 原始访问标识将代您的用户获取这些对象。如果您的用户尝试使用 Amazon S3 URL 访问对象，他们将会被拒绝访问。原始访问标识有权访问 Amazon S3 存储桶中的对象，但用户无权访问这些对象。

Note

要创建原始访问标识，您必须使用 CloudFront 控制台或 CloudFront API 的 2009-09-09 版本或更高版本。

要确保您的用户只使用 CloudFront URL 访问您的对象，而不考虑这些 URL 是否签名，请执行以下任务：

1. 创建原始访问标识并将其添加到您的分配中。有关更多信息，请参阅 创建 CloudFront 原始访问标识并将其添加到您的分配中 (p. 95)。

   Note

   您也可在创建分配时创建原始访问标识并将其添加到您的分配中。

2. 更改对您的 Amazon S3 存储桶或存储桶中对象的权限，以便只有原始访问标识具有读取权限（或读取和下载权限）。

   有关更多信息，请参阅 向原始访问标识授予读取 Amazon S3 存储桶中对象的权限 (p. 97)。

创建 CloudFront 原始访问标识并将其添加到您的分配中

一个 AWS 账户最多可拥有 100 个 CloudFront 原始访问标识。但是，您可将一个原始访问标识添加到任意多项分配中，因此一个原始访问标识通常就足够了。

如果在您创建分配时未创建原始访问标识并将其添加到您的分配中，您现在可使用 CloudFront 控制台或 CloudFront API 创建并添加一个：

- 如果您使用 CloudFront 控制台：您可在创建原始访问标识的同时将其添加到您的分配中。有关更多信息，请参阅 使用 CloudFront 控制台创建原始访问标识并将其添加到您的分配中 (p. 96)。

- 如果您使用 CloudFront API：您可先创建一个原始访问标识，然后再将其添加到您的分配中。执行以下每个主题中的步骤：
  - 使用 CloudFront API 创建原始访问标识 (p. 97)
  - 使用 CloudFront API 将原始访问标识添加到您的分配中 (p. 97)

使用 CloudFront 控制台创建原始访问标识并将其添加到您的分配中

如果您在创建分配时未创建原始访问标识，请执行以下步骤。

使用 CloudFront 控制台创建 CloudFront 原始访问标识

1. 登录 AWS 管理控制台，并通过以下网址打开 Amazon CloudFront 控制台：https://console.aws.amazon.com/cloudfront/。
2. 单击您要将原始访问标识添加到其中的分配的 图标。
3. 切换到编辑模式：
   - Web 分配：单击 Origins (源) 选项卡，单击您想编辑的源，然后单击 Edit (编辑)。您只能为 Origin Type (源类型) 为 S3 Origin (S3 源) 的源创建原始访问标识。
   - RTMP 分配：单击 Edit (编辑)。
4. 对于 Restrict Bucket Access (限制存储桶访问)，请单击 Yes (是)。
5. 如果您已经具有您想使用的原始访问标识，请单击 Use an Existing Identity (使用现有标识)。然后在 Your Identities (您的标识) 列表中选择该身份。

   Note

   如果您已经有原始访问标识，我们建议您重用该标识以便简化维护。

   如果您想创建一个标识，请单击 Create a New Identity (创建新标识)。然后在 Comment (备注) 字段中输入对该标识的描述。
6. 如果您希望 CloudFront 自动向原始访问标识提供读取 Origin Domain Name (源域名) 中指定的 Amazon S3 存储桶中对象的权限，请单击 Yes, Update Bucket Policy (是，更新存储桶策略)。

   Important

   如果您单击 Yes, Update Bucket Policy (是，更新存储桶策略)，CloudFront 将更新存储桶权限，以向指定的原始访问标识授予读取存储桶中对象的权限。但是，CloudFront 不会删除现有权限。如果用户目前有权使用 Amazon S3 URL 访问存储桶中的对象，则他们在 CloudFront 更新存储桶权限后将仍具有该权限。要查看或删除现有存储桶权限，请使用 Amazon S3 提供的方法。有关更多信息，请参阅 向原始访问标识授予读取 Amazon S3 存储桶中对象的权限 (p. 97)。

   如果您希望手动更新对 Amazon S3 存储桶的权限，请单击 No, I Will Update Permissions (否，我将更新权限)。
7. 单击 Yes, Edit (是，编辑)。
8. 如果您要向 Web 分配添加原始访问标识，并且您有多个源，请重复执行第 3 步到第 7 步（如适用）。

使用 CloudFront API 创建原始访问标识

如果您已经具有一个原始访问标识并要重用它，而不是另外创建一个，请跳到使用 CloudFront API 将原始访问标识添加到您的分配中 (p. 97)。

要使用 CloudFront API 创建 CloudFront 原始访问标识，请采用 `POST Origin Access Identity` API 操作。所得到的响应将包含用于新原始访问标识的一个 `Id` 和一个 `S3CanonicalUserId`。记下这些值，因为在该过程中的后面部分会用到它们：

- Id 元素：您需要使用 `Id` 元素的值将原始访问 ID 与您的分配关联起来。
- S3CanonicalUserId 元素：当您向 CloudFront 授予对 Amazon S3 存储桶或对象的访问权限时，您需要用到 `S3CanonicalUserId` 元素的值。

有关 `POST Origin Access Identity API` 操作的更多信息，请访问 *Amazon CloudFront API 参考*中的 POST 原始访问标识。关于您可对原始访问标识执行的其他操作的列表，请访问同样位于 *Amazon CloudFront API 参考*中的针对原始访问标识的操作。

使用 CloudFront API 将原始访问标识添加到您的分配中

您可使用 CloudFront API 将 CloudFront 原始访问标识添加到现有分配中，也可创建包含原始访问标识的新分配。在这两种情况下，都请包含 `OriginAccessIdentity` 元素。该元素包含您创建原始访问标识时 `POST Origin Access Identity` API 操作返回的 `Id` 元素的值。对于 Web 分配，请将 `OriginAccessIdentity` 元素添加到一个或多个源中。对于 RTMP 分配，请将 `OriginAccessIdentity` 元素添加到该分配中。

请参阅 *Amazon CloudFront API 参考*中的适用主题：

- 创建新的 Web 分配：POST 分配
- 更新现有的 Web 分配：PUT 分配配置
- 创建新的 RTMP 分配：POST 流分配
- 更新现有的 RTMP 分配：PUT 流分配配置

向原始访问标识授予读取 Amazon S3 存储桶中对象的权限

当您创建或更新分配时，您可添加一个原始访问标识并自动更新存储桶策略，以向原始访问标识提供访问存储桶的权限。另外，您也可选择手动更改存储桶策略或更改控制存储桶中各对象权限的 ACL。

无论使用哪种方法，您仍应检查存储桶的存储桶策略并检查对象的权限，以确保：

- CloudFront 可代使用 CloudFront URL 请求对象的用户访问存储桶中的对象。
- 用户不能使用 Amazon S3 URL 访问您的对象。

Caution

如果您将 CloudFront 配置为接受 CloudFront 支持的所有 HTTP 方法并将这些方法转发到 Amazon S3，请创建 CloudFront 原始访问标识，以限制对您的 Amazon S3 内容的访问，并向原始访问标识授予适用的许可。例如，如果因为您想使用 `PUT` 方法而将 CloudFront 配置为接受和转发这些方法，则必须将 Amazon S3 存储桶策略或 ACL 配置为适当处理 `DELETE` 请求，以便用户无法删除您不希望其删除的资源。

请注意以下几点：

- 您可能会发现更新 Amazon S3 存储桶策略比更新 ACL 更容易，因为您在将对象添加到存储桶时可以不更新权限。不过，使用 ACL 时您可以进行更细粒度的控制，因为您授予的是对每个对象的权限。
- 默认情况下，您的 Amazon S3 存储桶和其中的所有对象都是私有的 — 只有创建存储桶的 AWS 账户有权读写其中的对象。
- 如果您要向现有分配添加原始访问标识，请视情况需要修改存储桶策略或任何对象 ACL，以确保这些对象不可公开使用。
- 向一个或多个安全管理员账户授予额外的权限，以便您可继续更新 Amazon S3 存储桶的内容。

Important

您保存对 Amazon S3 权限的更改后，可能需经过短暂的延迟后更改才能生效。在更改生效之前，在您试图访问存储桶中的对象时可能会出现权限被拒错误。

更新 Amazon S3 存储桶策略

您可以使用 AWS Management Console或 Amazon S3 API 更新 Amazon S3 存储桶策略：

- 向 CloudFront 原始访问标识授予对存储桶的适用权限。

  要指定原始访问标识，请使用 CloudFront 控制台中 Origin Access Identity (原始访问标识) 页面上 Amazon S3 Canonical User ID (Amazon S3 规范用户 ID) 的值。如果您使用的是 CloudFront API，请使用您创建原始访问标识时返回的 `S3CanonicalUserId` 元素值。
- 拒绝您不希望其有权使用 Amazon S3 URL 进行访问的任何人。

有关更多信息，请访问 *Amazon Simple Storage Service 开发者指南*中的使用存储桶策略。

有关示例，请参阅同样位于 *Amazon Simple Storage Service 开发者指南*中的 Amazon S3 存储桶策略示例情形主题中的“使用规范 ID 向 CloudFront 原始标识授予权限”。

更新 Amazon S3 ACL

使用 AWS Management Console或 Amazon S3 API 更改 Amazon S3 ACL：

- 向 CloudFront 原始访问标识授予对 CloudFront 分配提供的每个对象的适用权限。

  要指定原始访问标识，请使用 CloudFront 控制台中 Origin Access Identity (原始访问标识) 页面上 Amazon S3 Canonical User ID (Amazon S3 规范用户 ID) 的值。如果您使用的是 CloudFront API，请使用您创建原始访问标识时返回的 `S3CanonicalUserId` 元素值。
- 拒绝您不希望其有权使用 Amazon S3 URL 进行访问的任何人。

如果另一个 AWS 账户向您的存储桶上传对象，则该账户将成为这些对象的拥有者。默认情况下，拥有存储桶中对象的账户是可授予对这些对象的权限的唯一账户。但是，拥有对象的 AWS 账户可将您也设为拥有者，从而使您可更改对对象的权限。

有关详细信息，请访问 *Amazon Simple Storage Service 开发者指南*中的使用 ACL。

您也可使用代码和 AWS 开发工具包之一更改 ACL。有关示例，请参阅使用 C# 和 .NET Framework 创建 URL 签名 (p. 123)中的可下载示例代码。

指定可创建签名 URL 的 AWS 账户（可信签署人）

Topics

要为 Amazon S3 存储桶中的对象创建签名 URL，您至少需要一个具有有效 CloudFront 密钥对的 AWS 账户。该账户称作可信签署人。可信签署人有两个用途：

- 您将可信签署人的 AWS 账号添加到您的分配中后，CloudFront 就会立即开始要求该用户使用签名 URL 访问 Amazon S3 存储桶中的对象。
- 当您创建签名 URL 时，请使用可信签署人的密钥对中的私钥来签署 URL 的一部分。当有人使用签名 URL 访问对象时，CloudFront 将比较该 URL 的已签名部分和未签名部分，以验证该 URL 是否未遭篡改。CloudFront 还会验证该 URL 是否有效，例如，验证该 URL 中的过期日期和时间是否还未到。

当您指定可信签署人时，还需间接指定要求使用签名 URL 的对象：

- Web 分配：将可信签署人添加到缓存行为。如果您的分配只有一项缓存行为，用户必须使用签名 URL 访问与该分配有关的任何对象。如果您创建了多项缓存行为，并将可信签署人添加到某些缓存行为而不是其他缓存行为，则您可以要求用户使用签名 URL 访问某些对象而不是其他对象。
- RTMP 分配：将可信签署人添加到分配中。在您将可信签署人添加到 RTMP 分配后，用户必须使用签名 URL 访问与该分配有关的任何对象。

Note

要为分配指定可信签署人，您必须使用 CloudFront 控制台或 CloudFront API 的 2009-09-09 版本或更高版本。

要指定可创建签名 URL 的账户并将这些账户添加到 CloudFront 分配中，请执行以下任务：

1. 决定您要使用哪些 AWS 账户作为可信签署人。大多数 CloudFront 客户都使用他们用于创建分配的账户。
2. 针对您在第 1 步中选择的每个账户，创建一个 CloudFront 密钥对。有关更多信息，请参阅 为可信签署人创建 CloudFront 密钥对 (p. 99)。
3. 如果您使用 .NET 或 Java 创建签名 URL，请重新设置 CloudFront 私钥的格式。有关更多信息，请参阅 重新设置 CloudFront 私钥的格式（仅适用于 .NET 和 Java）(p. 100)。
4. 在您为其创建签名 URL 的分配中，指定可信签署人的 AWS 账号。有关更多信息，请参阅 将可信签署人添加到您的分配中 (p. 100)。
5. 可选：验证 CloudFront 是否能识别出您的可信签署人具有有效的 CloudFront 密钥对。当可信签署人具有有效的 CloudFront 密钥对时，CloudFront 会识别出它们为有效的可信签署人。有关更多信息，请参阅 验证可信签署人是否有效（可选）(p. 102)。

为可信签署人创建 CloudFront 密钥对

您用于创建 CloudFront 签名 URL 的每个 AWS 账户（可信签署人）必须创建自己的 CloudFront 密钥对，并且密钥对必须有效。请注意以下几点：

- 目前不允许 IAM 用户创建 CloudFront 密钥对，因此，您不能使用 IAM 用户作为可信签署人。
- 您不能用 Amazon EC2 密钥对替换 CloudFront 密钥对。在创建 CloudFront 签名 URL 时，您需将可信签署人密钥对的密钥对 ID 包含在 URL 中。Amazon EC2 不提供密钥对 ID。

要帮助保护您的应用程序，我们建议您每 90 天或更频繁地更改 CloudFront 密钥对。有关更多信息，请参阅 轮换 CloudFront 密钥对 (p. 103)。

为可信签署人创建 CloudFront 密钥对

1. 登录 AWS Management Console，然后打开“Your Security Credentials (您的安全证书)”页面，网址为 https://console.aws.amazon.com/iam/home?#security_credential。
2. 展开 CloudFront Key Pairs (CloudFront 密钥对)。
3. 单击 Create New Key Pair (创建新的密钥对)。
4. 在 Create Key Pair (创建密钥对) 对话框中，单击 Download Private Key File (下载私钥文件)。
5. 在 Opening <filename> (打开 <文件名>) 对话框中，接受 Save File (保存文件) 的默认值，然后单击 OK (确定)，以下载并保存 CloudFront 密钥对的私钥。

   Important

   将 CloudFront 密钥对的私钥保存在安全的位置，并设置对文件的权限，以便只有所需的管理员用户可读取它。如果有人获得您的私钥，他们便可以生成有效的签名 URL 并下载您的内容。您无法再次获得该私钥，因此，如果您丢失了或删除了它，您必须创建一个新的 CloudFront 密钥对。

6. 记录您的密钥对的密钥对 ID。在您创建签名 URL 时会用到它。

重新设置 CloudFront 私钥的格式（仅适用于 .NET 和 Java）

如果您使用 .NET 或 Java 创建签名 URL，则您不能以默认的 .pem 格式使用密钥对中的私钥来创建签名：

- .NET 框架：将私钥转换成 .NET Framework 使用的 XML 格式。可使用多种工具达到此目的。对于使用 C# 和 .NET Framework 创建 URL 签名 (p. 123)中的示例，我们使用 .NET 2.0 OpenSSL 公匙和私钥分析器，网址为http://www.jensign.com/opensslkey/opensslkey.cs。
- Java：将私钥转换成 DER 格式。为此，您可使用 OpenSSL：

  ```
  $ openssl pkcs8 -topk8 -nocrypt -in origin.pem -inform PEM -out new.der
  -outform DER
  ```

  要确保编码器正常工作，请将 Bouncy Castle Java 加密 API 的 jar 添加到您的项目中，然后添加 Bouncy Castle 提供程序。

将可信签署人添加到您的分配中

可信签署人是可为分配创建签名 URL 的 AWS 账户。默认情况下，不允许任何账户为分配创建签名 URL，甚至是创建该分配的账户也不例外。要指定您想用作可信签署人的 AWS 账户，请将相应的账户添加到您的分配中：

- Web 分配：可信签署人与缓存行为有关。这样，您就可以针对同一分配中的某些对象要求使用签名 URL，而对于其他对象则不作要求。可信签署人只可为与对应缓存行为有关的对象创建签名 URL。例如，如果您有两个分别与两项不同缓存行为关联的不同可信签署人，则这两个可信签署人均不能为与对方缓存行为有关的对象创建签名 URL。

- RTMP 分配：可信签署人与分配有关。在您将可信签署人添加到 RTMP 分配后，用户必须使用签名 URL 访问与该分配有关的任何对象。

您必须小心定义缓存行为的路径模式，以确保 CloudFront 视情况要求或不要求使用签名 URL。

Caution

请小心定义路径模式及它们的顺序，以避免误向最终用户提供对内容的访问权。例如，假设某一请求与两项缓存行为的路径模式匹配。第一项缓存行为不要求使用签名 URL，而第二项缓存行为则要求使用签名 URL。这种情况下最终用户不需使用签名 URL 即可访问对象，因为 CloudFront 处理的是与第一个匹配项有关的缓存行为。

有关路径模式的更多信息，请参阅路径模式 (p. 33)。

Caution

如果您要更新您已经用于分发内容的分配，请在准备好开始为对象生成签名 URL 后仅添加可信签署人。在您将可信签署人添加到 RTMP 分配后，用户必须使用签名 URL 访问与该分配有关的任何对象。在您将可信签署人添加到 Web 分配的缓存行为后，用户必须使用签名 URL 才能访问与该缓存行为有关的对象。

可信签署人的最大数量取决于分配的类型：

- Web 分配：每项缓存行为最多五个可信签署人
- RTMP 分配：这种分配最多五个可信签署人

您可以使用 CloudFront 控制台或 CloudFront API 将可信签署人添加到您的分配中。请参阅适用主题：

- 使用 CloudFront 控制台将可信签署人添加到分配中 (p. 101)
- 使用 CloudFront API 将可信签署人添加到分配中 (p. 102)

使用 CloudFront 控制台将可信签署人添加到分配中

使用 CloudFront 控制台将可信签署人添加到分配中

1. 如果您想只使用创建分配的 AWS 账户作为可信签署人，请跳到第 2 步。

   如果您想使用其他 AWS 账户，请获取每个账户的 AWS 账号：

   a. 使用您想用作可信签署人的账户登录 AWS Management Console，网址为 https://console.aws.amazon.com/console/home。
   b. 在控制台的右上角，单击与该账户关联的名称，然后单击 My Account (我的账户)。
   c. 记下出现在页面右上角的账号。
   d. 单击 Sign Out (退出)。
   e. 针对您想用作可信签署人的其他账户重复执行步骤 a 到步骤 d。

2. 打开 Amazon CloudFront 控制台（网址为 https://console.aws.amazon.com/cloudfront/），并使用您在创建要将可信签署人添加到的分配时所用的账户登录。
3. 单击您要将原始访问标识添加到其中的分配的  图标。
4. 切换到编辑模式：

   - Web 分配：单击 Behaviors (行为) 选项卡，单击您想编辑的行为，然后单击 Edit (编辑)。

- RTMP 分配：单击 Edit (编辑)。

5. 对于 Restrict Viewer Access (Use Signed URLs) (限制查看器访问 (使用签名 URL))，请单击 Yes (是)。
6. 对于 Trusted Signers (可信签署人)，请选中相应的复选框：
   - Self (本身)：如果您想使用当前账户（您用于创建分配的账户），请选中该复选框。
   - Specify Accounts (指定账户)：如果您想使用其他 AWS 账户，请选中该复选框。
7. 如果您选中了 Specify Accounts (指定账户) 复选框，请在 AWS Account Number (AWS 账号) 字段中输入 AWS 账号（带或不带连字符）。每行请输入一个账号。
8. 单击 Yes, Edit (是，编辑)。
9. 如果您要将可信签署人添加到 Web 分配，并且您有多项缓存行为，请重复执行第 4 步到第 8 步（如适用）。

使用 CloudFront API 将可信签署人添加到分配中

您可使用 CloudFront API 将可信签署人的 AWS 账号添加到现有分配中，也可创建包含可信签署人的新分配。在这两种情况下，都需在 `TrustedSigners` 元素中指定适用的值。对于 Web 分配，请将 `TrustedSigners` 元素添加到一项或多项缓存行为中。对于 RTMP 分配，请将 `TrustedSigners` 元素添加到该分配中。

请参阅 *Amazon CloudFront API 参考*中的适用主题：

- 创建新的 Web 分配：POST 分配
- 更新现有的 Web 分配：PUT 分配配置
- 创建新的 RTMP 分配：POST 流分配
- 更新现有的 RTMP 分配：PUT 流分配配置

验证可信签署人是否有效（可选）

在您将可信签署人添加到分配后，您可能需要验证该签署人是否有效。可信签署人必须满足以下条件才有效：

- AWS 账户必须至少具有一个有效的密钥对。如果您轮换密钥对，该账户将暂时有两个有效的密钥对：旧密钥对和新密钥对。
- CloudFront 必须知道有效的密钥对。在创建密钥对后，可能需要经过一段很短的时间，CloudFront 才会知道该密钥对存在。

Note

要显示某项分配的有效可信签署人列表，您当前必须使用 CloudFront API。在 CloudFront 控制台中不提供有效可信签署人的列表。

使用 CloudFront API 验证可信签署人是否有效

要确定哪些可信签署人具有有效密钥对（属于有效可信签署人），您可获取分配并检查 `ActiveTrustedSigners` 元素中的值。该元素列出了分配指定为可信签署人的每个账户的 AWS 账号。如果可信签署人具有一个或多个有效 CloudFront 密钥对，`ActiveTrustedSigners` 元素也列出密钥对 ID。有关更多信息，请参阅 *Amazon CloudFront API 参考*中的适用主题：

- Web 分配：GET 分配
- RTMP 分配：GET 流分配

轮换 CloudFront 密钥对

AWS 建议您每 90 天轮换（更换）一次您的有效 CloudFront 密钥对。要轮换您用于创建签名 URL 的 CloudFront 密钥对，而不使尚未过期的 URL 失效，请执行以下任务：

1. 为您用于创建签名 URL 的每个账户创建新的密钥对。有关更多信息，请参阅 为可信签署人创建 CloudFront 密钥对 (p. 99)。
2. 验证 CloudFront 是否知道新密钥对。有关更多信息，请参阅 验证可信签署人是否有效（可选） (p. 102)。
3. 更新您的应用程序，以使用新密钥对中的私钥创建签名。
4. 确认您使用新私钥签署的签名 URL 起作用。
5. 等至使用旧 CloudFront 密钥对签署的签名 URL 中的过期日期已过为止。
6. 将旧 CloudFront 密钥对更改为 Inactive (无效)：
   a. 访问您的安全证书页面。
   b. 展开 CloudFront Key Pairs (CloudFront 密钥对)。
   c. 针对适用的密钥对，单击 Make Inactive (设为无效)。
7. 重新确认您使用新私钥签署的签名 URL 起作用。
8. 删除旧 CloudFront 密钥对：
   a. 访问您的安全证书页面。
   b. 展开 CloudFront Key Pairs (CloudFront 密钥对)。
   c. 针对适用的密钥对，单击 Delete (删除)。
9. 从保存旧私钥的位置将旧私钥删除。

签名 URL 概述

Topics

既然您已将 CloudFront 配置为要求最终用户使用专用的签名 URL 访问您的对象，那么您就可以创建签名 URL（手动或以编程方式）并将其分发给您希望有权访问这些对象的用户。

签名 URL 还包含可帮助您控制对内容的访问的其他信息，例如，过期日期和时间。这些其他信息出现在基于标准策略或自定义策略的策略声明中。标准策略和自定义策略之间的差别将在接下来的两节中予以说明。

Note

对于同一项分配，您可使用标准策略创建一些签名 URL，再使用自定义策略创建一些签名 URL。

使用标准策略创建的签名 URL

如果您希望达到以下目的，请使用标准策略创建 CloudFront 签名 URL：

- 限制访问单个对象。
- 仅根据您希望用户停止访问的日期和时间来控制对您对象的访问。

您不需要将策略包含在 URL 内，因此标准策略产生的 URL 较短。您需要包含要据以对策略执行散列和签署操作的签名。

当用户使用您采用标准策略创建的签名 URL 请求对象时，CloudFront 将根据该 URL 中的信息重新构造标准策略声明。CloudFront 随后会将重新构造的策略声明与签名中的策略声明进行比较，以确定是否允许最终用户访问内容。如果两种声明不完全匹配，CloudFront 将拒绝访问内容。

有关使用标准策略创建签名 URL 的信息，请参阅使用标准策略创建签名 URL (p. 107)。

使用自定义策略创建的签名 URL

如果您想达到以下目的，请使用自定义策略创建 CloudFront 签名 URL：

- 限制访问一个或多个对象，例如，年度报告目录中的所有 .pdf 文件。这使得您可以对多个对象的签名 URL 使用单一的策略声明。
- 根据以下条件控制对您对象的访问：
  - 您希望用户停止拥有访问权限的日期和时间。
  - 您希望用户开始拥有访问权限的日期和时间。（可选）
  - 你希望有权访问对象的用户的 IP 地址或 IP 地址范围。（可选）

对于您使用自定义策略创建的签名 URL，您可以像对使用标准策略创建的签名 URL 一样，对策略进行散列和签署。此外，您将策略的 Base64 编码版本包含在 URL 内，这样就提供了 URL 安全压缩。因此，使用自定义策略所产生的 URL 要长于使用标准策略产生的 URL。

当用户使用您采用标准策略创建的签名 URL 请求对象时，CloudFront 会将签名 URL 中的自定义策略声明与签名中的策略声明进行比较，以确定是否允许最终用户访问内容。如果两种声明不完全匹配，CloudFront 将拒绝访问内容。

有关使用自定义策略创建签名 URL 的信息，请参阅使用自定义策略创建签名 URL (p. 112)。

签名 URL 的组成部分

CloudFront 签名 URL 包含以下组成部分。

基本 URL

基本 URL 是您在您不使用签名 URL 时将用于访问对象的 CloudFront URL，其中包括您自己的查询字符串参数（如有）。有关适用于 Web 分配的 URL 格式的更多信息，请参阅CloudFront 对象的 URL 格式 (p. 57)。

以下示例显示了您为 Web 分配指定的值。

- 以下 CloudFront URL 用于 Web 分配中的对象（使用 CloudFront 域名）。请注意，`image.jpg` 在 `images` 目录中。URL 中对象的路径必须与您 HTTP 服务器或您 Amazon S3 存储桶中对象的路径匹配。

  ```
  http://d111111abcdef8.cloudfront.net/images/image.jpg
  ```
- 以下 CloudFront URL 包含查询字符串：

  ```
  http://d111111abcdef8.cloudfront.net/images/image.jpg?size=large
  ```
- 以下 CloudFront URL 用于 Web 分配中的对象。两者都使用备用域名，第二个包括查询字符串：

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

  ```
  http://www.example.com/images/image.jpg?color=red
  ```
- 以下 CloudFront URL 用于使用备用域名和 HTTPS 协议的 Web 分配中的对象：

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

对于 RTMP 分配，以下示例用于采用两种不同视频格式（MP4 和 FLV）的对象：

- MP4：`mp4:sydney-vacation.mp4`
- FLV：`sydney-vacation`
- FLV：`sydney-vacation.flv`

Note

对于 .flv 文件，是否包括 `.flv` 文件扩展名取决于您的播放器。要提供 MP3 音频文件或 H.264/MPEG-4 视频文件，您可能需要用 `mp3:` 或 `mp4:` 作为文件名的前缀。某些媒体播放器可配置为自动添加前缀。媒体播放器可能也会要求您指定不带文件扩展名的文件名（例如，sydney-vacation，而不是 sydney-vacation.mp4）。

过期日期和时间（仅限标准策略）

您希望 URL 停止允许访问对象的日期和时间，采用 Unix 时间格式（按秒计）和协调通用时间 (UTC)。例如，UTC 时间 2013 年 1 月 1 日上午 10 点转换成 Unix 时间格式后将为 1357034400。有关 UTC 的更多信息，请参阅 *RFC 3339，互联网上的日期和时间：时间戳*，网址为 http://tools.ietf.org/html/rfc3339。

如果您使用的是自定义策略，您仍可指定您希望签名 URL 停止允许访问对象的日期和时间，但该日期和时间包含在策略声明内，而不是作为单独的查询字符串参数出现在签名 URL 中。

策略声明

策略声明是采用 JSON 格式的文本字符串，决定着签名 URL 的特征。对于标准策略和自定义策略，策略声明包含对象的 URL（对于 Web 分配）或流名称（对于 RTMP 分配），以及过期日期和时间。请注意以下几点：

- 标准策略：您无需将策略声明包括在签名 URL 中 — 只需创建策略声明，以便您可对它进行散列及签署，并将该签名包括在 URL 中。请参阅 签名 (p. 106)。
- 自定义策略：您还可选择包括 URL 生效的日期和时间和/或可访问对象的 IP 地址或 IP 地址范围。然后，您需要对策略声明进行 Base64 编码，并将编码后的策略声明包括在签名 URL 内。

签名

签名是策略声明经散列和签署后的版本。

创建签名 URL 的 AWS 账户的 CloudFront 密钥对 ID

CloudFront 密钥对 ID 告诉 CloudFront 要使用哪个公钥来验证签名 URL。CloudFront 将比较签名中的信息与策略声明中的信息，以验证该 URL 是否未遭篡改。

您包括在 CloudFront 签名 URL 中的密钥对 ID 必须是您可信签署人之一的有效密钥对的 ID。有关更多信息，请参阅 指定可创建签名 URL 的 AWS 账户（可信签署人）(p. 99)。

如果您在轮换 CloudFront 密钥对时使密钥对无效，并且您是以编程方式生成签名 URL 的，您必须更新您的应用程序，使其使用可信签署人之一的新有效密钥对。如果您是以手动方式生成签名 URL 的，您必须创建新的签名 URL。有关轮换密钥对的更多信息，请参阅轮换 CloudFront 密钥对 (p. 103)。

CloudFront 何时检查签名 URL 中的过期日期和时间？

CloudFront 何时检查签名 URL 中的过期日期和时间来确定该 URL 是否仍有效，取决于该 URL 是用于 Web 分配还是用于 RTMP 分配：

- Web 分配：CloudFront 在发出 HTTP 请求时检查签名 URL 中的过期日期和时间。如果客户端刚好在过期时间前的那一刻开始下载大型对象，那么即使在下载过程中到了过期时间，该下载也应完成。如果 TCP 连接断开，并且客户端试图在过期时间到期后重新开始下载，则下载将会失败。

  如果客户端使用范围 GET 来获取较小的对象，在过期时间到期后发出的任何 GET 请求都将失败。有关范围 GET 的更多信息，请参阅CloudFront 如何处理对象的部分请求（范围 GET）(p. 74)。
- RTMP 分配：CloudFront 在播放事件开始时检查签名 URL 中的过期时间。如果客户端在过期时间到期之前开始播放媒体文件，CloudFront 将允许播放完整个媒体文件。不过，根据媒体播放器的不同，暂停并重新开始播放可能触发另一个播放事件。跳到媒体文件中另一个位置将触发另一个播放事件。如果在过期时间到期后发生后续播放事件，CloudFront 不会提供媒体文件。

使用标准策略创建签名 URL

使用标准策略创建签名 URL

1. 如果您要使用 .NET 和 Java 创建签名 URL，并且您尚未将密钥对私钥的格式从默认的 .pem 格式重新设置为与 .NET 和 Java 兼容的格式，那么现在就请开始设置。有关更多信息，请参阅 重新设置 CloudFront 私钥的格式（仅适用于 .NET 和 Java）(p. 100)。
2. 按指定顺序将下列值连接在一起并删除各部分之间的空格。您可能需要在应用程序代码的字符串中包含转义字符。所有值的类型均为 String。每一部分都有一个数字编号 (1)，后面紧跟两个示例。

1	*Base URL for the object* 这是您在不使用签名 URL 时用于访问对象的 URL，例如： • Web 分配：`http://d111111abcdef8.cloudfront.net/images/image.jpg` • RTMP 分配： `videos/mediafile.flv`
2	? ? 表示查询字符串参数遵循基本 URL 格式。即使您没有自己的任何查询字符串参数，也要包含 ?。
3	*Your query string parameters, if any*& 此值为可选项。如果您想添加自己的查询字符串参数，例如： `color=red&size=medium` 那么请在 ? 之后（请参阅 2）、`Expires` 参数之前添加这些参数。 **Important** 您的参数不能命名为 `Expires`、`Signature` 或 `Key-Pair-Id`。 如果您添加自己的参数，请在每个参数后追加 &，包括最后一个参数。
4	Expires=*date and time in Unix time format (in seconds) and Coordinated Universal Time (UTC)* 请以 Unix 时间格式和协调通用时间 (UTC) 指定过期日期和时间。例如，UTC 时间 2013 年 1 月 1 日上午 10 点转换成 Unix 时间格式后将为 1357034400。有关 UTC 的更多信息，请参阅 *RFC 3339，互联网上的日期和时间：时间戳*，网址为 http://tools.ietf.org/html/rfc3339。
5	&Signature=*hashed and signed version of the policy statement* 策略声明经散列和签署后的版本。有关更多信息，请参阅 为标准策略创建签名 (p. 108)。

6	&Key-Pair-Id=*active CloudFront key pair Id for the key pair that you are using to generate the signature* 有效 CloudFront 密钥对的 ID，例如 APKA9ONS7QCOWEXAMPLE： • Web 分配：密钥对必须与作为适用缓存行为可信签署人之一的 AWS 账户关联。 • RTMP 分配：密钥对必须与作为分配可信签署人之一的 AWS 账户相关联。 有关更多信息，请参阅 指定可创建签名 URL 的 AWS 账户（可信签署人）(p. 99)。

Web 分配的签名 URL 示例：

1 http://d111111abcdef8.cloudfront.net/image.jpg 2 ? 3 color=red&size=medium&
4 Expires=1357034400 5 &Signature=nitfHRCrtziwO2HwPfWw~yYDhUF5EwRunQA-j19DzZr
vDh6hQ73lDx~-ar3UocvvRQVw6EkC~GdpGQyyOSKQim-TxAnW7d8F5Kkai9HVx0FIu-
5jcQb0UEmatEXAMPLE3ReXySpLSMj0yCd3ZAB4UcBCAqEijkytL6f3fVYNGQI6
6 &Key-Pair-Id=APKA9ONS7QCOWEXAMPLE

RTMP 分配的签名 URL 示例：

1 videos/mediafile.flv 2 ? 3 color=red&size=medium& 4 Expires=1357034400
5 &Signature=nitfHRCrtziwO2HwPfWw~yYDhUF5EwRunQA-j19DzZr
vDh6hQ73lDx~-ar3UocvvRQVw6EkC~GdpGQyyOSKQim-TxAnW7d8F5Kkai9HVx0FIu-
5jcQb0UEmatEXAMPLE3ReXySpLSMj0yCd3ZAB4UcBCAqEijkytL6f3fVYNGQI6
6 &Key-Pair-Id=APKA9ONS7QCOWEXAMPLE

为标准策略创建签名

要为使用标准策略的签名 URL 创建签名，请执行以下两个步骤：

- 第一步是创建策略声明，下面紧接着就会介绍。
- 第二步是对策略声明进行散列和签署。该步骤有两个版本。您选择的版本取决于分配类型（是 Web 还是 RTMP）；对于 RTMP 分配，还取决于您使用的媒体播放器（是 Adobe Flash 播放器还是其他媒体播放器）。请使用第一步后的链接作为指导来选择第二步的适用版本。

对于使用标准策略的签名 URL，您无需像对待使用自定义策略的签名 URL 那样将策略声明包含在 URL 内。

有关其他信息以及如何对策略声明进行散列、签署和编码的示例，请参阅：

- 使用 Linux 命令和 OpenSSL 进行 Base64 编码和加密 (p. 119)
- 为签名 URL 创建签名的代码和示例 (p. 120)
- 用于配置私有内容的工具 (p. 251)

为使用标准策略的签名 URL 创建策略声明

1. 使用以下 JSON 格式以及 UTF-8 字符编码构建策略声明。请完全按照指定包括所有标点符号及其他文本值。

```
{"Statement":[{"Resource":"base URL or stream
name","Condition":{"DateLessThan":{"AWS:EpochTime":ending date and time in
Unix time format and UTC}}}]}
```

对于 Resource 和 DateLessThan，请指定以下值：

Resource：您指定的值取决于您是为 Web 分配还是为 RTMP 分配创建签名 URL：

- Web 分配：包含查询字符串（如果有）但不包含 CloudFront Expires、Signature 和 Key-Pair-Id 参数的基本 URL，例如：

  ```
  http://d111111abcdef8.cloudfront.net/images/horizon.jpg?size=large&license=yes
  ```

  请注意以下几点：
  - 值必须以 http://、https:// 或 http*:// 开头。
  - 如果您没有查询字符串参数，请省略问号。
  - 您可以使用与零个或更多个字符匹配的通配符 (*)，或者与字符串中任意位置的一个字符完全匹配的通配符 (?)。例如，下面的值：

    ```
    http*://d111111abcdef8.cloudfront.net/*game_download.zip*
    ```

    将包括（仅作示例）以下对象：

    ```
    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
    ```
  - 如果您在 URL 中指定备用域名（别名记录），您必须在引用网页或应用程序中的对象时指定备用域名。切勿指定对象的 Amazon S3 URL。
- RTMP 分配：仅包含流名称。例如，如果流视频的完整 URL 为：

  ```
  rtmp://s5c39gqb8ow64r.cloudfront.net/videos/mp3_name.mp3
  ```

  那么请对 Resource 使用以下值：

  ```
  videos/mp3_name
  ```

  切勿包含诸如 mp3: 或 mp4: 等前缀。此外，根据您所使用的播放器，您可能必须在 Resource 值中省略文件扩展名。例如，您可能需要使用 sydney-vacation 而不是 sydney-vacation.flv。

DateLessThan：采用 Unix 时间格式（按秒计）和协调通用时间 (UTC) 格式的 URL 过期日期和时间。例如，UTC 时间 2013 年 1 月 1 日上午 10 点转换成 Unix 时间格式后将为 1357034400。有关 UTC 的更多信息，请参阅 *RFC 3339，互联网上的日期和时间：时间戳*，网址为 http://tools.ietf.org/html/rfc3339。

该值必须与签名 URL 中的 Expires 查询字符串参数值相匹配。切勿用引号将该值引起来。

有关更多信息，请参阅 CloudFront 何时检查签名 URL 中的过期日期和时间？ (p. 106)。

示例

当您在签名 URL 中使用以下示例标准策略时，最终用户在 UTC 时间 2013 年 1 月 1 日上午 10 点前都可以访问对象 http://d111111abcdef8.cloudfront.net/horizon.jpg：

```
{"Statement":[{"Resource":"http://d111111abcdef8.cloudfront.net/hori
zon.jpg?size=large&license=yes","Condition":{"DateLessThan":{"AWS:Epoch
Time":1357034400}}}]}
```

如果您复制并粘贴此示例，请用您自己的值替换 URL 和过期时间。

2. 请删除策略声明中的所有空格。您可能需要在应用程序代码的字符串中包含转义字符。

执行适用的步骤，以便为您的签名 URL 创建签名：

- 选项 1：通过使用标准策略为 Web 分配或 RTMP 分配（不使用 Adobe Flash 播放器）创建签名 (p. 110)
- 选项 2：通过使用标准策略为 RTMP 分配（使用 Adobe Flash 播放器）创建签名 (p. 110)

选项 1：通过使用标准策略为 Web 分配或 RTMP 分配（不使用 Adobe Flash 播放器）创建签名

1. 使用 SHA-1 散列函数对您在为使用标准策略的签名 URL 创建策略声明 (p. 108) 步骤中创建的策略声明进行散列及签署。对于散列函数所需要的私钥，使用与适用的有效可信签署人有关的私钥。

   Note

   您用于对策略声明进行散列及签署的方法取决于您的编程语言和平台。有关示例代码，请参阅为签名 URL 创建签名的代码和示例 (p. 120)。

2. 删除散列及签署后的字符串中的空格。
3. 对字符串进行 Base64 编码。
4. 用有效的字符替换 URL 查询字符串中的无效字符。下表列出了无效和有效字符。

要被替换的无效字符	要替换成的有效字符
+	- （连字符）
=	_（下划线）
/	~（波浪字符）

5. 将所得到的值追加在签名 URL 中的 &Signature= 之后，然后返回使用标准策略创建签名 URL (p. 107)，以完成连接签名 URL 各个组成部分的操作。

选项 2：通过使用标准策略为 RTMP 分配（使用 Adobe Flash 播放器）创建签名

1. 使用 SHA-1 散列函数对您在为使用标准策略的签名 URL 创建策略声明 (p. 108) 步骤中创建的策略声明进行散列及签署。对于散列函数所需要的私钥，使用与适用的有效可信签署人有关的私钥。

   Note

   您用于对策略声明进行散列及签署的方法取决于您的编程语言和平台。有关示例代码，请参阅为签名 URL 创建签名的代码和示例 (p. 120)。

2. 删除散列及签署后的字符串中的空格。

   如果您正在使用 Adobe Flash 播放器且从网页中传入了流名称，请继续执行第 3 步。

如果您正在使用 Adobe Flash 播放器并且当前未从网页中传入流名称，请跳过本过程的其余步骤。例如，如果您编写您自己的用于从 Adobe Flash .swf 文件内获取流名称的播放器，请跳过本过程的其余步骤。

3. 对字符串进行 Base64 编码。
4. 用有效的字符替换 URL 查询字符串中的无效字符。下表列出了无效和有效字符。

要被替换的无效字符	要替换成的有效字符
+	- (连字符)
=	_ (下划线)
/	~ (波浪字符)

5. Adobe Flash 播放器的一些版本要求您对字符 ?、= 和 & 进行 URL 编码。有关您的 Adobe Flash 播放器版本是否要求进行这种字符替换的信息，请参阅 Adobe 网站。

   如果您的 Flash 版本不要求对这些字符进行 URL 编码，请跳到第 6 步。

   如果您的 Flash 版本要求对这些字符进行 URL 编码，根据下表中的指示替换他们。(您在上一步中已经替换了 =。)

要被替换的无效字符	要替换成的 URL 编码
?	%3F
&	%26

6. 将所得到的值追加在签名 URL 中的 &Signature= 之后，然后返回使用标准策略创建签名 URL (p. 107)，以完成连接签名 URL 各个组成部分的操作。

使用自定义策略创建签名 URL

Topics

要使用自定义策略创建签名 URL，请执行以下步骤。

使用自定义策略创建签名 URL

1. 如果您要使用 .NET 和 Java 创建签名 URL，并且您尚未将密钥对私钥的格式从默认的 .pem 格式重新设置为与 .NET 和 Java 兼容的格式，那么现在就请开始设置。有关更多信息，请参阅 重新设置 CloudFront 私钥的格式（仅适用于 .NET 和 Java） (p. 100)。
2. 按指定顺序将下列值连接在一起并删除各部分之间的空格。您可能需要在应用程序代码的字符串中包含转义字符。所有值的类型均为 String。每一部分都有一个数字编号 (1)，后面紧跟两个示例。

1	*Base URL for the object* 这是您在不使用签名 URL 时用于访问对象的 URL，例如： • Web 分配：`http://d111111abcdef8.cloudfront.net/images/image.jpg` • RTMP 分配： `videos/mediafile.flv`
2	? ? 表示查询字符串参数遵循基本 URL 格式。即使您没有自己的任何查询字符串参数，也要包含 ?。
3	*Your query string parameters, if any*& 此值为可选项。如果您想添加自己的查询字符串参数，例如： `color=red&size=medium` 那么请在 ?（请参阅 2）后、`Policy` 参数之前添加这些参数。 **Important** 您的参数不能命名为 `Policy`、`Signature` 或 `Key-Pair-Id`。 如果您添加自己的参数，请在每个参数后追加 &，包括最后一个参数。
4	Policy=*policy statement* JSON 格式的策略声明，删除了其中的空格。有关更多信息，请参阅 为自定义策略创建策略声明 (p. 113)。
5	&Signature=*hashed and signed version of the policy statement* 策略声明经散列和签署后的版本。有关更多信息，请参阅 为自定义策略创建签名 (p. 117)。

6	&Key-Pair-Id=*active CloudFront key pair Id for the key pair that you are using to sign the policy statement* 有效 CloudFront 密钥对的 ID，例如 APKA9ONS7QCOWEXAMPLE： • Web 分配：密钥对必须与作为适用缓存行为可信签署人之一的 AWS 账户关联。 • RTMP 分配：密钥对必须与作为分配可信签署人之一的 AWS 账户相关联。 有关更多信息，请参阅 指定可创建签名 URL 的 AWS 账户（可信签署人）(p. 99)。

Web 分配的签名 URL 示例：

1 http://d111111abcdef8.cloudfront.net/image.jpg 2 ? 3 color=red&size=medium&
4 Policy=eyANCiAgICEXAMPLEW1lbnQiOiBbeyANCiAgICAgICJSZXNvdXJjZSI6Imh0dHA
6Ly9kemJlc3FtN3VuMW0wLmNsb3VkZnJvbnQubmV0L2RlbW8ucGhwIiwgDQogICAgICAiQ
29uZGl0aW9uIjp7IA0KICAgICAgICAgIkIwQWRkcmVzcyI6eyJBV1M6U291cmNlSXAiOiI
yMDcuMTcxLjE4MC4xMDEvMzIifSwNCiAgICAgICAgICJEYXRlR3JlYXRlclRoYW4iOnsiQ
VdTOkVwb2NoVGltZSI6MTI5Njg2MDE3Nn0sDQogICAgICAgICAiRGF0ZUxlc3NUaGFuIjp
7IkFXUzpFcG9jaFRpbWUiOjEyOTY4NjAyMjZ9DQogICAgICAgICB9IA0KICAgfV0gDQo
5 &Signature=nitfHRCrtziwO2HwPfWw~yYDhUF5EwRunQA-j19DzZrvDh6hQ73lDx~
-ar3UocvvRQVw6EkC~GdpGQyyOSKQim-TxAnW7d8F5Kkai9HVx0Flu-5jcQb0UEmat
EXAMPLE3ReXySpLSMj0yCd3ZAB4UcBCAqEijkytL6f3fVYNGQI6
6 &Key-Pair-Id=APKA9ONS7QCOWEXAMPLE

RTMP 分配的签名 URL 示例：

1 videos/mediafile.flv 2 ? 3 color=red&size=medium&
4 Policy=eyANCiAgICEXAMPLEW1lbnQiOiBbeyANCiAgICAgICJSZXNvdXJjZSI6Imh0dHA
6Ly9kemJlc3FtN3VuMW0wLmNsb3VkZnJvbnQubmV0L2RlbW8ucGhwIiwgDQogICAgICAiQ
29uZGl0aW9uIjp7IA0KICAgICAgICAgIkIwQWRkcmVzcyI6eyJBV1M6U291cmNlSXAiOiI
yMDcuMTcxLjE4MC4xMDEvMzIifSwNCiAgICAgICAgICJEYXRlR3JlYXRlclRoYW4iOnsiQ
VdTOkVwb2NoVGltZSI6MTI5Njg2MDE3Nn0sDQogICAgICAgICAiRGF0ZUxlc3NUaGFuIjp
7IkFXUzpFcG9jaFRpbWUiOjEyOTY4NjAyMjZ9DQogICAgICAgICB9IA0KICAgfV0gDQo
5 &Signature=nitfHRCrtziwO2HwPfWw~yYDhUF5EwRunQA-j19DzZrvDh6hQ73lDx~
-ar3UocvvRQVw6EkC~GdpGQyyOSKQim-TxAnW7d8F5Kkai9HVx0Flu-5jcQb0UEmat
EXAMPLE3ReXySpLSMj0yCd3ZAB4UcBCAqEijkytL6f3fVYNGQI6
6 &Key-Pair-Id=APKA9ONS7QCOWEXAMPLE

为自定义策略创建策略声明

要为自定义策略创建策略声明，请执行以下步骤。有关以各种方式控制对对象的访问的一些示例策略声明，请参阅自定义策略的示例策略声明 (p. 115)。

为使用自定义策略的签名 URL 创建策略声明

1. 使用以下 JSON 格式构建策略声明。

```
{
   "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"}
      }
   }]
}
```

请注意以下几点：

- 请使用 UTF-8 字符编码。
- 请完全按照指定包含所有标点符号和参数名称。不接受参数名称的缩写。
- `Condition` 部分中参数的顺序无关紧要。
- 有关 `Resource`、`DateLessThan`、`DateGreaterThan` 和 `IpAddress` 值的信息，请参阅此步骤后的说明。

2. 请删除策略声明中的所有空格。您可能需要在应用程序代码的字符串中包含转义字符。
3. 请将所得到的值追加在签名 URL 中 `Policy=` 之后。
4. 通过对策略声明进行散列、签署和 Base64 编码，为签名 URL 创建签名。有关更多信息，请参阅 为自定义策略创建签名 (p. 117)。

Resource

- Web 分配（可选，但建议使用）：包含查询字符串（如果有）但不包含 CloudFront `Policy`、`Signature` 和 `Key-Pair-Id` 参数的基本 URL，例如：

  ```
  http://d111111abcdef8.cloudfront.net/images/horizon.jpg?size=large&license=yes
  ```

  Caution

  如果您省略 Web 分配的 Resource 参数，最终用户将可以访问与您用于创建签名 URL 的密钥对相关的任何分配的所有关联对象。

  请注意以下几点：

  - 值必须以 `http://`、`https://` 或 `*` 开头。
  - 如果您没有查询字符串参数，请省略问号。
  - 您可以使用与零个或更多个字符匹配的通配符 (*)，或者与字符串中任意位置的一个字符完全匹配的通配符 (?)。例如，下面的值：

    ```
    http*://d111111abcdef8.cloudfront.net/*game_download.zip*
    ```

    将包括（仅作示例）以下对象：

    ```
    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
    ```

  - 如果您在 URL 中指定备用域名（别名记录），您必须在引用网页或应用程序中的对象时指定备用域名。切勿指定对象的 Amazon S3 URL。
- RTMP 分配：仅包含流名称。例如，如果流视频的完整 URL 为：

  `rtmp://s5c39gqb8ow64r.cloudfront.net/videos/mp3_name.mp3`

  那么请对 `Resource` 使用以下值：

  `videos/mp3_name`

  切勿包含诸如 `mp3:` 或 `mp4:` 等前缀。此外，根据您所使用的播放器，您可能必须在 `Resource` 值中省略文件扩展名。例如，您可能需要使用 `sydney-vacation` 而不是 `sydney-vacation.flv`。

DateLessThan

采用 Unix 时间格式（按秒计）和协调通用时间 (UTC) 格式的 URL 过期日期和时间。切勿用引号将该值引起来。有关 UTC 的更多信息，请参阅 *RFC 3339，互联网上的日期和时间：时间戳*，网址为 http://tools.ietf.org/html/rfc3339。

例如，UTC 时间 2013 年 1 月 1 日上午 10 点转换成 Unix 时间格式后将为 1357034400。

这是在 `Condition` 部分中唯一的必需参数。CloudFront 需要使用此值来防止用户对您的私有内容有永久访问权。

有关更多信息，请参阅 CloudFront 何时检查签名 URL 中的过期日期和时间？ (p. 106)。

DateGreaterThan（可选）

采用 Unix 时间格式（按秒计）和协调通用时间 (UTC) 格式的 URL 可选开始日期和时间。不允许用户在指定的日期和时间之前访问对象。切勿用引号将该值引起来。

IpAddress（可选）

发出 GET 请求的客户端的 IP 地址。要允许任何 IP 地址访问对象，请省略此参数。

IP 地址范围必须采用标准的 IPv4 CIDR 格式（例如，`10.52.176.0/24`）。有关更多信息，请访问 *RFC 4632，无类域间路由 (CIDR)：互联网地址分配和聚合计划*，网址为 http://tools.ietf.org/html/rfc4632。

您指定为条件指定单个值。例如，如果客户端的 IP 地址在两个独立范围中的一个范围内，您就无法将策略设置为允许访问。

自定义策略的示例策略声明

以下示例策略声明说明了如何控制对具体对象、目录中的所有对象或与密钥对 ID 有关的所有对象的访问。这些示例还说明了如何控制来自单个 IP 地址或 IP 地址范围的访问，以及如何防止用户在指定日期和时间后使用签名 URL。

如果您复制并粘贴其中的任何示例，请删除所有空格，使用自己的值替换相应的值，并在右花括号 (}) 后包含一个换行符。

示例策略声明：从 IP 地址范围访问一个对象

例如，下面是签名 URL 中的一项自定义策略，它规定最终用户在 UTC 时间 2013 年 1 月 1 日上午 10 点前，都可以从 `192.0.2.0/24` 范围内的 IP 地址访问对象 `http://d111111abcdef8.cloudfront.net/game_download.zip`：

```
{
   "Statement": [{
      "Resource":"http://d111111abcdef8.cloudfront.net/game_download.zip",
      "Condition":{
         "IpAddress":{"AWS:SourceIp":"192.0.2.0/24"},
         "DateLessThan":{"AWS:EpochTime":1357034400}
      }
   }]
}
```

示例策略声明：从 IP 地址范围访问目录中的所有对象

以下示例自定义策略允许您为 `training` 目录中的任何对象创建签名 URL，`Resource` 参数中的 * 通配符指示了这一点。最终用户在 UTC 时间 2013 年 1 月 1 日上午 10 点前，都可从 `192.0.2.0/24` 范围内的 IP 地址访问对象：

```
{
   "Statement": [{
      "Resource":"http://d111111abcdef8.cloudfront.net/training/*",
      "Condition":{
         "IpAddress":{"AWS:SourceIp":"192.0.2.0/24"},
         "DateLessThan":{"AWS:EpochTime":1357034400}
      }
   }]
}
```

您在其中使用此策略的每个签名 URL 都包含一个指定了具体对象的基本 URL，例如：

`http://d111111abcdef8.cloudfront.net/training/orientation.pdf`

示例策略声明：从一个 IP 地址访问与密钥对 ID 有关的所有对象

以下示例自定义策略允许您为与任何分配有关的任何对象创建签名 URL，`Resource` 参数中的 * 通配符指示了这一点。最终用户必须使用 IP 地址 `192.0.2.10/32`。（用 CIDR 表示法指定的值 `192.0.2.10/32` 引用单个 IP 地址 `192.0.2.10`。）对象仅在 UTC 时间 2013 年 1 月 1 日上午 10 点到 UTC 时间 2013 年 1 月 2 日上午 10 点期间可用：

```
{
   "Statement": [{
      "Resource":"http://*",
      "Condition":{
         "IpAddress":{"AWS:SourceIp":"192.0.2.10/32"},
         "DateGreaterThan":{"AWS:EpochTime":1357034400},
         "DateLessThan":{"AWS:EpochTime":1357120800}
      }
   }]
}
```

您在其中使用此策略的每个签名 URL 均包含指定具体 CloudFront 分配中具体对象的基本 URL，例如：

`http://d111111abcdef8.cloudfront.net/training/orientation.pdf`

签名 URL 还包含一个密钥对 ID，该 ID 必须与您在基本 URL 中指定的分配 (d111111abcdef8.cloudfront.net) 中的可信签署人相关。

为自定义策略创建签名

使用自定义策略的签名 URL 的签名是策略声明经散列、签署及 Base64 编码后的版本。要为自定义策略创建签名，请执行适用的步骤。您选择的版本取决于分配类型（是 Web 还是 RTMP）；对于 RTMP 分配，还取决于您使用的媒体播放器（是 Adobe Flash 播放器还是其他媒体播放器）：

- 选项 1：通过使用自定义策略为 Web 分配或 RTMP 分配（不使用 Adobe Flash 播放器）创建签名 (p. 117)
- 选项 2：通过使用自定义策略为 RTMP 分配（使用 Adobe Flash 播放器）创建签名 (p. 117)

有关其他信息以及如何对策略声明进行散列、签署和编码的示例，请参阅：

- 使用 Linux 命令和 OpenSSL 进行 Base64 编码和加密 (p. 119)
- 为签名 URL 创建签名的代码和示例 (p. 120)
- 用于配置私有内容的工具 (p. 251)

选项 1：通过使用自定义策略为 Web 分配或 RTMP 分配（不使用 Adobe Flash 播放器）创建签名

1. 使用 SHA-1 散列函数对您在为使用自定义策略的签名 URL 创建策略声明 (p. 113) 步骤中创建的策略声明进行散列及签署。对于散列函数所需要的私钥，使用与适用的有效可信签署人有关的私钥。

   Note

   您用于对策略声明进行散列及签署的方法取决于您的编程语言和平台。有关示例代码，请参阅为签名 URL 创建签名的代码和示例 (p. 120)。

2. 删除散列及签署后的字符串中的空格。
3. 对字符串进行 Base64 编码。
4. 用有效的字符替换 URL 查询字符串中的无效字符。下表列出了无效和有效字符。

要被替换的无效字符	要替换成的有效字符
+	- （连字符）
=	_ （下划线）
/	~ （波浪字符）

5. 将所得到的值追加在签名 URL 中的 &Signature= 之后，然后返回使用自定义策略创建签名 URL (p. 112)，以完成连接签名 URL 各个组成部分的操作。

选项 2：通过使用自定义策略为 RTMP 分配（使用 Adobe Flash 播放器）创建签名

1. 使用 SHA-1 散列函数对您在为使用自定义策略的签名 URL 创建策略声明 (p. 113) 步骤中创建的策略声明进行散列及签署。对于散列函数所需要的私钥，使用与适用的有效可信签署人有关的私钥。

   Note

   您用于对策略声明进行散列及签署的方法取决于您的编程语言和平台。有关示例代码，请参阅为签名 URL 创建签名的代码和示例 (p. 120)。

2. 删除散列及签署后的字符串中的空格。

如果从网页中传入了流名称，请继续执行第 3 步。

如果未从网页中传入流名称，请跳过本过程的其余步骤。例如，如果您编写您自己的用于从 Adobe Flash .swf 文件内获取流名称的播放器，请跳过本过程的其余步骤。

3. 对字符串进行 Base64 编码。
4. 用有效的字符替换 URL 查询字符串中的无效字符。下表列出了无效和有效字符。

要被替换的无效字符	要替换成的有效字符
+	- (连字符)
=	_ (下划线)
/	~ (波浪字符)

5. Adobe Flash 播放器的一些版本要求您对字符 ?、= 和 & 进行 URL 编码。有关您的 Adobe Flash 播放器版本是否要求进行这种字符替换的信息，请参阅 Adobe 网站。

   如果您的 Adobe Flash 播放器版本不要求您对字符 ?、= 和 & 进行 URL 编码，请跳到第 6 步。

   如果您的 Adobe Flash 播放器要求对这些字符进行 URL 编码，请按下表中的指示替换它们。(您在上一步中已经替换了 =。)

要被替换的无效字符	要替换成的 URL 编码
?	%3F
&	%26

6. 将所得到的值追加在签名 URL 中的 &Signature= 之后，然后返回使用自定义策略创建签名 URL (p. 112)，以完成连接签名 URL 各个组成部分的操作。

使用 Linux 命令和 OpenSSL 进行 Base64 编码和加密

您可以使用 Linux 命令行命令和 OpenSSL 来执行以下操作：

- 对策略声明进行 Base64 编码，并用有效字符替换无效字符。
- 将策略声明转换为签名。

有关 OpenSSL 的信息，请访问 http://www.openssl.org/。

对策略声明进行 Base64 编码

以下 Linux 命令对策略声明（在文件策略中）进行 Base64 编码并用有效字符替换 URL 查询字符串参数中的无效字符：

```
(1) cat policy | (2) openssl base64 | (3) tr '+=/' '-_~'
```

其中：

(1)	`cat` 用于将策略文件发送至 openssl。
(2)	OpenSSL 对文件进行 Base64 编码。
(3)	`tr` 用有效的字符替换 URL 查询字符串参数中的无效字符。

将策略声明转换为签名。

以下 Linux 命令对策略声明进行散列、签署和 Base64 编码，以创建一个签名：

```
(1) cat policy | (2) openssl sha1 -sign private-key.pem | (3) openssl base64
| (4) tr '+=/' '-_~'
```

其中：

(1)	`cat` 用于将经过 Base64 编码的 `policy` 文件发送至 OpenSSL。
(2)	OpenSSL 使用 SHA-1 对文件进行散列并使用私钥文件 `private-key.pem` 对其进行签署。
(3)	OpenSSL 对经过散列和签署的策略声明进行 Base64 编码。
(4)	`tr` 用有效的字符替换 URL 查询字符串参数中的无效字符。

Note

请从产生的签名中删除所有空格（如有）。

如需以多种编程语言演示如何创建签名的代码示例，请参阅为签名 URL 创建签名的代码和示例 (p. 120)。

为签名 URL 创建签名的代码和示例

本节包括演示如何为签名 URL 创建签名的可下载应用程序示例。这些示例有 Perl、PHP、C# 和 Java 语言版本。所有示例均可用于创建签名 URL。Perl 脚本在 Linux/Mac 平台上运行。PHP 示例在运行 PHP 的任何服务器上都可正常工作。C# 示例使用 .NET Framework。

Topics

使用 Perl 创建 URL 签名

Perl 脚本根据命令行参数为私有内容创建签名，这些命令行参数指定了 CloudFront URL、签署人私钥的路径、密钥 ID 以及 URL 的过期日期。该工具还可对经过签名的 URL 进行解码。要获得 `cfsign.pl` 工具，请访问 Amazon CloudFront 签名 URL 帮助程序工具。

Note

创建 URL 签名只是使用签名 URL 提供私有内容过程的一个环节。有关整个过程的更多信息，请参阅私有内容的工作原理 (p. 92)。

以下示例说明了可通过何种方式使用 `cfsign.pl` 来创建 RTMP 分配签名。

```
$ cfsign.pl --action encode --stream example/video.mp4 --private-key
   /path/to/my-private-key.pem --key-pair-id PK12345EXAMPLE --expires 1265838202
```

该工具根据命令行参数生成策略声明。它所生成的签名是经 SHA1 散列后的策略声明。

以下是采用 Base64 编码的流名称的一个示例。

```
mp4:example/video.mp4%3FPolicy%3DewogICJTdGF0ZW1lbnQiOlt7CiAgICAgICJSZXNvdXJjZSI
6ImRyciIsCiAgICAgICJDb25kaXRpb24iOnsKICAgICAgICAiSXBBZGRyZXNzIjp7IkFXUzpTb3VyY2V
JcCI6IjAuMC4wLjAvMCJ9LAogICAgICAgICJEYXRlTGVzc1RoYW4iOnsiQVdTOkVwb2NoVGltZSI6MjE
0NTkxNjgwMH0KICAgICAgfQogICAgEXAMPLE_%26Signature%3DewtHqEXK~68tsZt-eOFnZKGwTf2a
JlbKhXkK5SSiVqcG9pieCRV3xTEPtc29OzeXlsDvRycOM2WK0cXzcyYZhpl9tv2796ihHiCTAwIHQ8yP
17Af4nWtOLIZHoH6wkR3tU1cQHs8R1d-g-SlZGjNBXr~J2MbaJzm8i6EXAMPLE_%26Key-Pair-Id%3
DPK12345EXAMPLE
```

此签名会验证请求的身份，以便对私有内容 `example/video.mp4` 进行流式处理。如果您正在使用 Adobe Flash 播放器并且使用 JavaScript 从网页传入了流名称，您必须对签名进行 Base64 编码，并用有效的字符（-、_ 和 ~）分别替换 URL 请求参数中的无效字符（+、=、/）。如果未从网页传入流名称，您不必对签名进行 Base64 编码。例如，如果您编写自己的播放器并从 Adobe Flash .swf 文件内获取流名称，您将不需要对签名进行 Base64 编码。

以下示例对 CloudFront 采用 jwplayer。

```
<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>
```

当您从 Adobe Flash .swf 文件中检索流进行播放时，切勿对流名称进行 URL 编码，例如：

```
mp4:example/video.mp4?Policy=ewogICJTdGF0ZW1lbnQiOlt7CiAgICAgICJSZXNvdXJjZSI6ImR
yciIsCiAgICAgICJDb25kaXRpb24iOnsKICAgICAgICAiSXBBZGRyZXNzIjp7IkFXUzpTb3VyY2VJcCI
6IjAuMC4wLjAvMCJ9LAogICAgICAgICJEYXRlTGVzc1RoYW4iOnsiQVdTOkVwb2NoVGltZSI6MjE0NTk
xNjgwMH0KICAgICAgfQogICAgEXAMPLE_&Signature=ewtHqEXK~68tsZt-eOFnZKGwTf2aJlbKhXkK
5SSiVqcG9pieCRV3xTEPtc29OzeXlsDvRycOM2WK0cXzcyYZhpl9tv2796ihHiCTAwIHQ8yP17Af4nWt
OLIZHoH6wkR3tU1cQHs8R1d-g-SlZGjNBXr~J2MbaJzm8i6EXAMPLE_&Key-Pair-Id=PK12345
EXAMPLE
```

有关该工具的命令行开关和功能的更多信息，请参阅 Perl 源代码中的注释。

另请参阅

- 使用 PHP 创建 URL 签名 (p. 121)
- 使用 C# 和 .NET Framework 创建 URL 签名 (p. 123)
- 使用 Java 创建 URL 签名 (p. 131)
- 用于配置私有内容的工具 (p. 251)

使用 PHP 创建 URL 签名

运行 PHP 的任何 Web 服务器可使用 PHP 演示代码，来为私有的 CloudFront RTMP 分配创建策略声明和签名。该示例使用签名 URL 链接创建一个可正常运行的网页，用以通过 CloudFront 流式处理来播放视频流。要获得该示例，请下载采用 PHP 的视频流签名代码。

Note

创建 URL 签名只是使用签名 URL 提供私有内容过程的一个环节。有关整个过程的更多信息，请参阅私有内容的工作原理 (p. 92)。

在以下代码段中，`rsa_sha1_sign` 函数对策略声明进行散列和签署。所需的参数有：策略声明、包含签名的输出参数以及 AWS 账户或您指定的可信 AWS 账户的私钥。接下来，`url_safe_base64_encode` 函数会创建签名的 URL 安全版本。

Example 采用 PHP 进行的 RSA SHA1 散列

```
 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);
 }
```

以下代码构建了创建签名所需的*标准*策略声明。有关标准策略的更多信息，请参阅使用标准策略创建签名 URL (p. 107)。

Example 采用 PHP 的标准签名函数

```
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);
    }
```

以下代码构建了创建签名所需的*自定义* 策略声明。有关自定义策略的更多信息，请参阅使用自定义策略创建签名 URL (p. 112)。

Example 采用 PHP 的自定义签名函数

```
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);

    }
```

有关 SHA-1 的 OpenSSL 实现的更多信息，请参阅适用于 SSL/TLS 的开源工具包。

另请参阅

- 使用 Perl 创建 URL 签名 (p. 120)
- 使用 C# 和 .NET Framework 创建 URL 签名 (p. 123)
- 使用 Java 创建 URL 签名 (p. 131)
- 用于配置私有内容的工具 (p. 251)

使用 C# 和 .NET Framework 创建 URL 签名

本节中的 C# 示例实现了一个示例应用程序，该程序演示了如何使用标准和自定义策略声明为 CloudFront 私有分配创建签名。这些示例中包含了基于 AWS .NET 开发工具包的实用程序函数，这些函数可在 .NET 应用程序中使用。

Note

创建 URL 签名只是使用签名 URL 提供私有内容过程的一个环节。有关整个过程的更多信息，请参阅私有内容的工作原理 (p. 92)。

要下载代码，请访问采用 C# 的签名代码。

要在 .NET Framework 中使用 AWS 账户/安全功能提供的 RSA 密钥，您必须将 AWS 提供的 .pem 文件转换成 .NET Framework 使用的 XML 格式。.NET 2.0 OpenSSL 公匙和私钥分析器提供的 OpenSSL 公匙和私钥分析器将执行这种转换。

转换之后，RSA 私钥文件采用以下格式：

Example 采用 XML .NET Framework 格式的 RSA 私钥

```
<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>
```

以下 C# 代码通过以下方式创建使用标准策略的签名 URL：

- 创建策略声明。
- 使用 SHA1 对策略声明进行散列，并使用 RSA 以及您的 AWS 账户或您指定的可信 AWS 账户的私钥签署结果。
- 对经过散列和签署的策略声明进行 Base64 编码，并替换特殊字符，以使字符串可安全地用作 URL 请求参数。
- 将适用的值连接在一起。

有关完整的实现，请参阅采用 C# 的签名代码中的示例。

Example 采用 C# 的标准策略签名方法

```
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;
    }
}
```

以下 C# 代码通过以下方式创建使用自定义策略的签名 URL：

- 创建策略声明。
- 对策略声明进行 Base64 编码，并替换特殊字符，以使字符串可安全地用作 URL 请求参数。
- 使用 SHA1 对策略声明进行散列，并使用 RSA 以及您的 AWS 账户或您指定的可信 AWS 账户的私钥对结果进行加密。
- 对经过散列的策略声明进行 Base64 编码，并替换特殊字符，以使字符串可安全用作 URL 请求参数。
- 将适用的值连接在一起。

有关完整的实现，请参阅采用 C# 的签名代码中的示例。

Example 采用 C# 的自定义策略签名方法

```
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 用于生成签名的实用程序方法

以下方法从文件中获得策略声明并分析时间间隔以生成签名。

```
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();
   }
```

另请参阅

- 使用 Perl 创建 URL 签名 (p. 120)
- 使用 PHP 创建 URL 签名 (p. 121)
- 使用 Java 创建 URL 签名 (p. 131)
- 用于配置私有内容的工具 (p. 251)

使用 Java 创建 URL 签名

适用于 Amazon S3 和 CloudFront 的开源 Java 工具包提供有关使用 Java 开发 CloudFront 的示例代码和信息。有关私有分配的信息，请访问程序员指南：示例代码中的“私有分配”。

Note

创建 URL 签名只是使用签名 URL 提供私有内容过程的一个环节。有关整个过程的更多信息，请参阅私有内容的工作原理 (p. 92)。

以下方法来自适用于 Amazon S3 和 CloudFront 的 Java 开源工具包。您必须将 PEM 格式的私钥转换成 DER 格式以便 Java 实现可使用它。

Example Java 策略和签名加密方法

```
// 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);
```

另请参阅

- 使用 Perl 创建 URL 签名 (p. 120)
- 使用 PHP 创建 URL 签名 (p. 121)
- 使用 C# 和 .NET Framework 创建 URL 签名 (p. 123)
- 用于配置私有内容的工具 (p. 251)

使用 HTTPS 连接访问您的对象

Abstract

通过对 CloudFront 采用 HTTPS 连接来控制对您对象的访问。

Topics

对于 Web 分配，您可使用 HTTPS 请求来确保您的对象在 CloudFront 将其提供给查看器时处于加密状态；如果需要，还可以确保在 CloudFront 从您的源获取这些对象时它们也处于加密状态。当您创建 Web 分配时，您可以：

- 配置一项或多项 CloudFront 缓存行为，以要求查看器仅使用 HTTPS 协议访问 CloudFront 缓存中的对象。这样，您就可以针对某些对象要求使用 HTTPS，而对于其他对象则不作要求。
- 配置一个或多个 CloudFront 源，以要求 CloudFront 使用查看器用于请求对象的协议来从源获取对象。例如，当您使用这种 CloudFront 设置且查看器使用 HTTPS 从 CloudFront 请求对象时，CloudFront 也会使用 HTTPS 将请求转发给您的源。当您的源为 Amazon S3 存储桶时，这是默认设置，且不能更改。

如果您使用 HTTP 服务器作为您的源，或者您想在查看器与 CloudFront 之间以及在 CloudFront 与您的源之间都使用 HTTPS，则您必须在 HTTP 服务器上安装经第三方证书颁发机构（例如 VeriSign 或 DigiCert）签署的 SSL 证书。

Caution

如果原始服务器返回无效证书或自签名证书，或者原始服务器以错误顺序返回证书链，则 CloudFront 将中断 TCP 连接，返回 HTTP 错误码 503，并将 `X-Cache` 标头设置为 `Error from cloudfront`。

CloudFront 如何使用 HTTPS 连接

下面关于 CloudFront 如何使用 HTTPS 连接的示例假定满足以下条件：

- 您的 CloudFront 分配有一项缓存行为（默认缓存行为）和一个源。
- 您已将您的分配配置为在查看器与 CloudFront 之间以及在 CloudFront 与您的源之间使用 HTTPS。
- 您的源具有经第三方证书颁发机构签署的 SSL 证书。

无论您的原始服务器是 Amazon S3 存储桶还是 HTTP 服务器，该过程的工作方式都基本相同。

使用 HTTPS 提供对象的 CloudFront 过程

1. 查看器向 CloudFront 提交 HTTPS 请求。对此，查看器与 CloudFront 之间会进行一定的 SSL 协商。最后，查看器以加密格式提交请求。
2. 如果对象在 CloudFront 节点缓存中，CloudFront 会将该对象加密并将其返回给查看器，然后查看器对其进行解密。
3. 如果对象不在 CloudFront 缓存中，则 CloudFront 会与您的源一起执行 SSL 协商，并在协商完成后将请求以加密格式转发给您的源。
4. 您的源对请求进行解密，对请求的对象进行加密，然后将该对象返回给 CloudFront。
5. CloudFront 对对象进行解密，并对其重新加密，然后将其转发给查看器。CloudFront 还会将该对象保存在节点缓存中，以供下次请求它时使用。
6. 查看器对对象进行解密。

如何要求在查看器、CloudFront 和您的源之间采用 HTTPS 进行通信

您可将 CloudFront 配置为要求在查看器和 CloudFront 之间或者在 CloudFront 和您的源之间采用 HTTPS 进行通信（可选）。

Note

要确保对象在从源到 CloudFront 节点缓存以及从节点缓存到查看器的传输过程中都处于加密状态，请仅使用 HTTPS。如果您曾经将 CloudFront 配置为使用 HTTP 从您的源获取对象，则 CloudFront 会将对象添加到节点缓存，并继续将它们提供给查看器，直至对象过期或者您删除或替换它们为止。有关删除或替换分配中的对象的更多信息，请参阅在分配中添加、删除或替换对象 (p. 62)。

如果要使用备用域名（例如 `example.com`），而不是使用 CloudFront 为分配指定的域名，也请参阅使用备用域名和 HTTPS (p. 136)。

要求在查看器、CloudFront 和您的源之间使用 HTTPS 进行通信

1. 如果您是使用 HTTP 服务器作为您的源，或者您还没有为服务器安装 SSL 证书，请从第三方证书颁发机构（如 VeriSign 或 DigiCert）获得 SSL 证书并加以安装。

   Note

   如果您使用的是 Amazon S3 存储桶，则 Amazon S3 会提供 SSL 证书。

   有关获得和安装 SSL 证书的更多信息，请参考 HTTP 服务器软件的文档和第三方证书颁发机构的文档。
2. 如果要求查看器在与 CloudFront 通信时使用 HTTPS，请在分配中创建或更新一项或多项缓存行为，并使其采用以下设置：

- CloudFront 控制台：对于 Viewer Protocol Policy (查看器协议策略)，请指定 HTTPS Only (仅限 HTTPS)。
- CloudFront API：对于 `ViewerProtocolPolicy`，请指定 `https-only`。

有关使用 CloudFront 控制台更新 Web 分配的信息，请参阅列出、查看及更新 CloudFront 分配 (p. 55)。

有关使用 CloudFront API 更新 Web 分配的信息，请访问 *Amazon CloudFront API 参考*中的 PUT 分配配置。

3. *可选*：如果要求 CloudFront 在与您的源通信时使用 HTTPS，请在分配中创建或更新一个或多个源，并使其采用以下设置：

   - CloudFront 控制台：对于 Origin Protocol Policy (源协议策略)，请指定 Match Viewer (匹配查看器)。
   - CloudFront API：对于 `OriginProtocolPolicy`，请指定 `match-viewer`。

   当您的源是 Amazon S3 存储桶时，Match Viewer (匹配查看器) 是默认设置，且不能更改。
4. 请确认是否满足以下条件：

   - 每项缓存行为中的路径模式仅适用于您希望查看器使用 HTTPS 的请求。
   - 缓存行为以所需顺序列出。有关更多信息，请参阅路径模式 (p. 33)。
   - 缓存行为将请求传送到您已为其将 Origin Protocol Policy (源协议策略) 配置为 Match Viewer (匹配查看器) 的源（如适用）。
   - 源具有第三方证书颁发机构签署的有效证书。

使用备用域名和 HTTPS

Topics

默认情况下，可以在您的 URL 中使用您的 CloudFront 分配域名（例如 `https://d111111abcdef8.cloudfront.net/image.jpg`），从而通过 HTTPS 将您的内容传输到查看器。有关更多信息，请参阅如何要求在查看器、CloudFront 和您的源之间采用 HTTPS 进行通信 (p. 135)。

如果希望您的查看器使用 HTTPS 并想在对象的 URL 中使用您自己的域名（例如 `https://www.example.com/image.jpg`），则您还需要执行以下操作：

- 将您自己的 SSL 证书上传到 AWS Identity and Access Management (IAM) 证书存储区。
- 将 IAM 证书存储区中您的 SSL 证书与您的 CloudFront 分配关联。
- 将一个或多个备用域名添加到您的分配。
- 添加或更新 DNS 记录，以将 DNS 查询传送到您的 CloudFront 分配。

当您将 SSL 证书与已启用的分配关联时，将产生额外的费用。有关更多信息，请参阅 http://aws.amazon.com/cloudfront/pricing。

Important

将某证书添加到您的分配时，CloudFront 会立即将该证书传播到其所有节点。当新的节点可用时，CloudFront 也会将该证书传播到这些位置。您不能限制 CloudFront 只能将您的证书传播到具体节点。

有关在 CloudFront 中使用 SSL 证书的要求和限制

请注意针对您证书的以下要求：

- 您的证书必须由经认可的证书颁发机构颁发。不接受自签名证书。
- 您的证书必须采用 X.509 PEM 格式。
- 在 .pem 文件中，请在证书链中列出所有中间证书，从为您的域签署证书的证书颁发机构所颁发的证书开始。示例如下：

```
-----BEGIN CERTIFICATE-----
CA public key certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
Intermediate certificate 2
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
Intermediate certificate 1
-----END CERTIFICATE-----
```

  我们建议证书链不要包含根证书。
- 私有密钥必须与证书中的公有密钥匹配。它还必须是一个 PEM 格式的 RSA 私有密钥，其中 PEM 页眉为 `BEGIN RSA PRIVATE KEY`，脚注为 `END RSA PRIVATE KEY`。私有密钥无法使用密码加密。
- 您必须有使用和上传 SSL 证书的权限（包括获得颁发相应证书的证书颁发机构的许可）才能有权将证书上传至内容传输网络。
- SSL 证书中公钥的最大大小为 2048 位。有关如何确定公钥大小的信息，请参阅确定 SSL 证书中公钥的大小 (p. 139)。
- CloudFront 支持所有类型的证书，包括经域验证的证书、扩展验证 (EV) 证书、高可靠性证书、通配符证书 (`*.example.com`)、使用者备用名称 (SAN) 证书（`example.com` 和 `example.net`）等。
- 对于您上传和在 CloudFront 中使用的 SSL 证书，由您负责监控证书过期日期和续订证书。

此外，请注意有关在 CloudFront 中使用 SSL 证书的以下限制：

- 最多可以将一个 SSL 证书与每个 CloudFront 分配关联。
- 默认情况下，如果您将一个 SSL 证书与您使用同一 AWS 账户创建的多项分配关联，则必须针对每项分配使用相同的证书。有关更多信息，请参阅对 HTTPS 采用备用域名 (p. 138)中的步骤 1。

  如果您已获准使用此功能，但需要增加您可以使用的自定义 SSL 证书的数量，请发送电子邮件到 cloudfront-ssl-request@amazon.com。在此电子邮件中，请注明您想使用多少个证书，并描述您的具体情况。请注意，这种限制的增加不会允许您将多个证书与同一分配关联。
- 对于每个 AWS 账户，最多可以将 10 个 SSL 证书上传至 IAM 证书存储区。要请求更高的限制，请访问请求增加 IAM 限制。
- 如果您要将同一个证书用于使用不同账户创建的多项 CloudFront 分配，则您必须针对每个 AWS 账户都将该证书上传到 IAM 证书存储区一次。

- 如果要对 CloudFront 和其他 AWS 服务使用同一个证书，则您必须上传该证书两次：为 CloudFront 上传一次，再为其他服务上传一次。有关如何上传 CloudFront 证书的信息，请参阅以下步骤。

对 HTTPS 采用备用域名

1. 申请 AWS 许可您针对您的 AWS 账户对 HTTPS 使用备用域名。我们将尽快更新您的账户。有关更多信息，请参阅 Amazon CloudFront 的自定义 SSL 证书。

   Important

   默认情况下，当您申请获得对 HTTPS 使用备用域名的许可时，AWS 会更新您的账户，以便您可以将两个自定义 SSL 证书与您的 CloudFront 分配关联。通常，您仅在有多项分配且您需要轮换证书时，才需临时使用第二个证书。如果您需要将两个或更多证书与您的分配永久关联，请指明您需要多少个证书，并在您的申请中描述具体情况。

2. 使用 AWS CLI 将您的 SSL 证书上传到 IAM 证书存储区。如果您尚未拥有证书，请参阅*使用 IAM* 中的创建、上传及删除服务器证书。

   如果您已有证书，请使用以下 AWS CLI 命令上传签名证书：

   ```
   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/
   ```

   请注意以下几点：

   - 您必须使用您用于创建 CloudFront 分配的同一个 AWS 账户，来将您的证书上传到 IAM 证书存储区。
   - 当您将证书上传至 IAM 时，`-path` 参数（证书路径）的值必须以 `/cloudfront/` 开头，例如 `/cloudfront/production/` 或 `/cloudfront/test/`。该路径还必须以 / 结尾。
   - 如果您计划使用 CloudFront 控制台来创建或更新您的分配，则您在 AWS CLI 中为 `--server-certificate-name` 参数指定的值是将显示在 CloudFront 控制台的 SSL Certificate (SSL 证书) 列表中的值。
   - 如果您计划使用 CloudFront API 来创建或更新您的分配，请记下 AWS CLI 返回的字母数字字符串，例如 `AS1A2M3P4L5E67SIIXR3J`。这是您将在 `IAMCertificateId` 元素中指定的值。您无需 IAM ARN（也由 CLI 返回）。

   有关 AWS CLI 的更多信息，请访问 *AWS Command Line Interface 用户指南* 及 *AWS Command Line Interface Reference*。

3. 将备用域名和您的 SSL 证书添加到您的分配，并添加或更新 DNS 记录。有关更多信息，请参阅 使用备用域名（别名记录）(p. 52)。

   Caution

   在将您的 SSL 证书与您的 CloudFront 分配关联之后，则您从所有分配中删除该证书并且分配的状态已变为 Deployed (已部署) 前，请勿从 IAM 证书存储区中删除该证书。

确定 SSL 证书中公钥的大小

如果您使用的是 CloudFront 备用域名和 HTTPS，则 SSL 证书中公钥的大小不得超过 2048 位。（这不是指公钥中的字符数。）您可以通过运行以下 OpenSSL 命令来确定公钥的大小：

```
openssl x509 -in path and filename of SSL certificate -text -noout
```

其中：

- `-in` 指定您的 SSL 证书的路径和文件名。
- `-text` 使 OpenSSL 以位为单位显示公钥的长度。
- `-noout` 阻止 OpenSSL 显示公钥。

示例输出：

```
Public-Key: (2048 bit)
```

轮换 SSL 证书

有时，您需要将一个 SSL 证书替换为其他证书，例如在过期日期临近时。具体过程取决于您是否已在同一 AWS 账户下将您的 SSL 证书与一个或多个 CloudFront 分配关联：

- SSL 证书与一项分配关联：您可以仅更新您的分配，将旧证书替换为新证书。有关更多信息，请参阅 列出、查看及更新 CloudFront 分配 (p. 55)。
- 在同一 AWS 账户下 SSL 证书与两项或更多分配关联：默认情况下，当您请求获得对 HTTPS 使用备用域名的许可时，在一个 AWS 账户下只能将两个 SSL 证书与 CloudFront 分配关联。通常，您仅在有多项分配且您需要轮换证书时，才会使用第二个证书。一个证书与您尚未更新的分配关联，另一个证书与您已更新的分配关联。请执行以下步骤。

Important

当您轮换证书时，可能会因使用第二个证书而产生按比例支付的额外费用。我们建议您及时更新您的分配，以最大程度地降低额外费用。

为两项或更多 CloudFront 分配轮换 SSL 证书

1. 如果您已经为您的账户关联 AWS 允许的最大数量的 SSL 证书，则需申请获得许可才能关联额外的证书。为此，请发送电子邮件至 cloudfront-ssl-request@amazon.com，并说明您要轮换证书。
2. 请一次更新一项分配，使其使用新证书。

   如果您在步骤 1 中向 AWS 提交了申请，请等至您收到通知称已更新 AWS 账户为止。

   有关更多信息，请参阅 列出、查看及更新 CloudFront 分配 (p. 55)。
3. （可选）更新您的所有 CloudFront 分配之后，您可以从 IAM 证书存储区中删除旧的证书。

Caution

在您将 SSL 证书从所有分配中删除并且您所更新的分配的状态变为 Deployed（已部署）前，请勿从 IAM 证书存储区中删除该证书。

从自定义 SSL 证书恢复为默认 CloudFront 证书

如果您已将 CloudFront 配置为使用自定义 SSL 证书，且您希望将您的配置更改为使用 CloudFront 的 SSL 证书，请执行以下步骤：

恢复为默认 CloudFront 证书

1. 采用所需的配置创建一项新的 CloudFront 分配。对于 SSL Certificate (SSL 证书)，请选择 Default CloudFront Certificate (*.cloudfront.net) (默认 CloudFront 证书 (*.cloudfront.net))。

   有关更多信息，请参阅 创建 Web 分配 (p. 19)。
2. 对于您要使用 CloudFront 分配的对象，请将您应用程序中的 URL 更新为使用 CloudFront 为新分配指定的域名。例如，将 `https://www.example.com/images/logo.png` 更改为 `https://d111111abcdef8.cloudfront.net/images/logo.png`。
3. 删除与自定义 SSL 证书关联的分配，或者更新分配，以将 SSL Certificate (SSL 证书) 的值更改为 Default CloudFront Certificate (*.cloudfront.net) (默认 CloudFront 证书 (*.cloudfront.net))。有关更多信息，请参阅 列出、查看及更新 CloudFront 分配 (p. 55)。

   Important

   在您完成此步骤之前，Amazon Web Services 会继续向您收取使用自定义 SSL 证书的费用。
4. *可选：*使用 AWS CLI 从 IAM 证书存储区中删除您的自定义 SSL 证书。这是您用于将自定义 SSL 证书添加到 IAM 证书存储区中的同一个应用程序：

   a. 运行 AWS CLI 命令 `list-signing-certificates`，以获取您要删除的证书的证书 ID。有关更多信息，请参阅 *AWS Command Line Interface Reference* 中的 list-signing-certificates。
   b. 运行 AWS CLI 命令 `delete-signing-certificate` 以删除该证书。有关更多信息，请参阅 *AWS Command Line Interface Reference* 中的 delete-signing-certificate。

HTTPS 连接的费用

您始终会产生与 HTTPS 请求相对应的额外费用。有关更多信息，请参阅 Amazon CloudFront 定价。

使用 IAM 控制对 CloudFront 资源的访问

Topics

Amazon CloudFront 与 AWS Identity and Access Management (IAM) 相集成，以便您可以创建您的 AWS 账户的用户，并且可以指定允许某一用户（或一组用户）在您的 AWS 账户中执行哪些 CloudFront 操作。通过创建描述用户或组权限的策略，您可以控制用户对 CloudFront 的访问。例如，您可以创建一个策略，只给予您组织中的某些用户以使用 `GetDistributionConfig` 的权限。然后，他们可使用操作来检索有关 CloudFront 分配的数据。

有关使用策略来设置 AWS 账户用户权限的更多信息，请转至*使用 AWS Identity and Access Management* 中的权限和策略。有关 IAM 的一般信息，请转至 AWS 网站上的 AWS Identity and Access Management。

Important

结合使用 Amazon CloudFront 和 IAM 并不会改变您使用 CloudFront 的方式。CloudFront 操作没有发生变化，并且没有与用户和访问控制相关的新 CloudFront 操作。

CloudFront 资源

当您编写一种策略来控制对 CloudFront 操作的访问时，应该使用星号 (*) 作为资源标记。这是因为您不能使用 IAM 来控制对特定 CloudFront 资源的访问。例如，您无法为用户提供对特定分配的访问权限。使用 IAM 授予的权限包括您用于 CloudFront 的所有资源。由于您不能指定要控制访问的资源，因此，您在 IAM 策略中没有可以使用 CloudFront 资源 ARN（Amazon 资源名称）。（有关在 IAM 中使用 ARN 的详细信息，请转至*使用 AWS Identity and Access Management* 中的 IAM 实体标识符。）

CloudFront 操作

在 IAM 策略中，您可以指定 CloudFront 提供的任何及所有 API 操作。操作名称必须使用小写字符串 `cloudfront:` 作为前缀。例如：`cloudfront:GetDistributionConfig`、`cloudfront:ListInvalidations` 或 `cloudfront:*`（针对所有 CloudFront 操作）。

下表列出了所有 CloudFront 操作的规范名称。当在 IAM 策略中指定 API 时，请使用这些规范名称。

Web 分配

Web 分配的 API 操作	规范名称
POST 分配	CreateDistribution
GET 分配	GetDistribution
GET 分配配置	GetDistributionConfig
PUT 分配配置	UpdateDistribution
GET 分配列表	ListDistributions
DELETE 分配	DeleteDistribution

RTMP 分配

RTMP 分配的 API 操作	规范名称
POST 流分配	CreateStreamingDistribution
GET 流分配	GetStreamingDistribution
GET 流分配配置	GetStreamingDistributionConfig
PUT 流分配配置	UpdateStreamingDistribution
GET 流分配列表	ListStreamingDistributions
DELETE 流分配	DeleteStreamingDistribution

失效

失效的 API 操作	规范名称
POST 失效	CreateInvalidation
GET 失效	GetInvalidation
GET 失效列表	ListInvalidations

原始访问标识

原始访问标识的 API 操作	规范名称
POST 原始访问标识	CreateCloudFrontOriginAccessIdentity
GET 原始访问标识	GetCloudFrontOriginAccessIdentity
GET 原始访问标识配置	GetCloudFrontOriginAccessIdentityConfig
PUT 原始访问标识配置	UpdateCloudFrontOriginAccessIdentity
GET 原始访问标识列表	ListCloudFrontOriginAccessIdentities
DELETE 原始访问标识	DeleteCloudFrontOriginAccessIdentity

策略密钥

策略密钥使您可以向策略添加条件，例如，请求日期或 IP 范围。CloudFront 可以实施 AWS 范围的策略密钥，但不实施其他策略。有关策略密钥的更多信息，请参阅*使用 AWS Identity and Access Management* 中元素描述部分的“条件”。

CloudFront 策略示例

本部分演示几个用于控制对 CloudFront 的用户访问的简单策略。

Note

今后，CloudFront 可能会添加一些新操作，这些操作应该会根据策略的既定目标，在逻辑上包含在以下策略之一中。

Example 1：允许对账户所拥有的全部资源进行组读写访问

此示例创建了一个附加到某一群组（例如，Developers (开发人员) 群组）的策略，用以给予该组对所有 CloudFront 资源的读写访问权限。

```
{
   "Version": "2012-10-17",
   "Statement":[{
      "Effect":"Allow",
      "Action":["cloudfront:*"],
      "Resource":"*"
      }
   ]
}
```

Example 2：允许对账户所拥有的全部资源进行组读取访问

此示例创建了一个附加到某一群组（例如，Finance(财务) 群组）的策略，用以给予该组对所有 CloudFront 资源的读取访问权限。

```
{
   "Version": "2012-10-17",
   "Statement":[{
      "Effect":"Allow",
      "Action":["cloudfront:Get*", "cloudfront:List*"],
      "Resource":"*"
      }
   ]
}
```

Example 3：允许对账户所拥有的全部分配进行组读写访问

此示例创建了一个附加到某一群组（例如，Ops(运营) 群组）的策略，用以给予该组对所有分配的组读写访问权限，但不允许访问失效或原始访问标识。

```
{
   "Version": "2012-10-17",
   "Statement":[{
      "Effect":"Allow",
      "Action":["cloudfront:*Distribution*"],
      "Resource":"*"
      }
   ]
}
```

Example 4：允许群组检索 CloudFront 分配数据，但前提是他们使用 SSL 提出请求。

此示例创建了一个附加到某一群组的策略，用以给予该组对所有 CloudFront 操作的访问权限，但条件是必须使用 SSL。

```
{
   "Version": "2012-10-17",
   "Statement":[{
      "Effect":"Allow",
      "Action":["cloudfront:*"],
      "Resource":"*",
      "Condition":{
         "Bool":{
            "aws:SecureTransport":"true"
            }
         }
      }
   ]
}
```

访问日志

Abstract

日志文件可提供有关用户请求的信息，这些请求针对的是您的应用程序和网站中的对象。

Topics

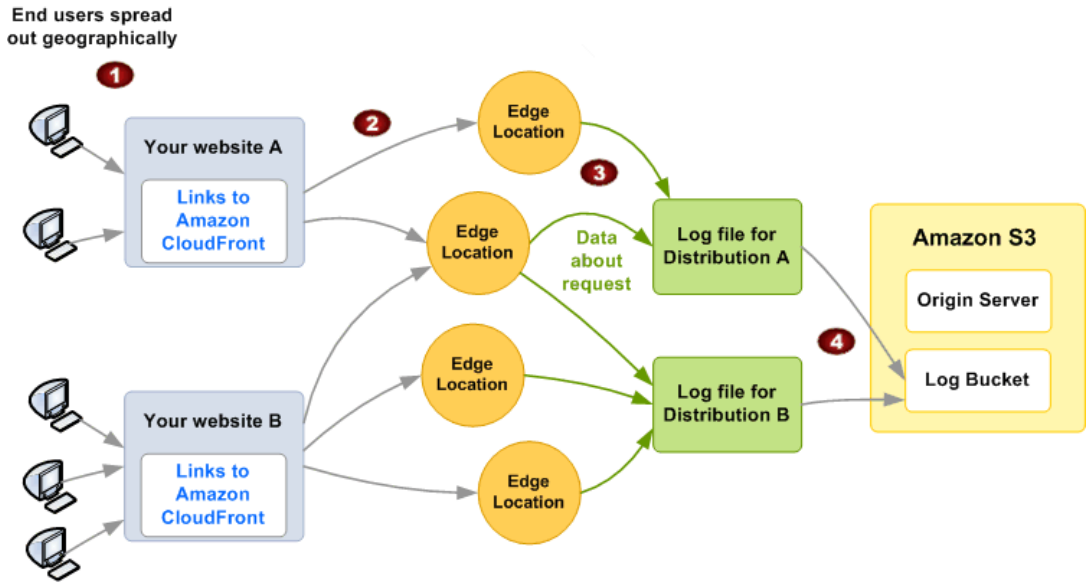
Amazon CloudFront 提供可选日志文件，其中包含有关用户请求的信息。本节说明了如何启用和禁用日志记录、日志文件内容以及 AWS 在您决定使用日志记录时如何收取费用。

Note

如果您使用自定义源，将需要创建 Amazon S3 存储桶，以在其中存储日志文件。

概述

您可以启用 CloudFront 以便按照分配将访问日志提供给您选择的 Amazon S3 存储桶。下图和表格说明了访问日志的基本流程。



访问日志的流程

1	您的最终用户使用您的应用程序或网站。 在此图中，您有两个不同的网站，A 和 B，每个都使用不同的 CloudFront 分配（分配 A 和分配 B）。
2	您的最终用户发送请求，CloudFront 将每个请求发送至适当的节点。
3	CloudFront 将每个请求的数据写入该分配特定的日志文件。在本例中，与分配 A 有关的请求信息将写入分配 A 的日志文件，与分配 B 有关的请求信息将写入分配 B 的日志文件。
4	CloudFront 定期将分配的日志文件放入您所选择的 Amazon S3 存储桶中，然后开始为分配写入新的日志文件。

日志文件中的每个条目分别提供某个请求的详细信息。有关文件格式的更多信息，请参阅日志文件格式 (p. 148)。

您可将分配的日志文件与您的原始服务器存储在同一 Amazon S3 存储桶中，也可将其存储在不同的存储桶中。（为了简化维护，我们建议您使用单独的存储桶。）您也可将多个分配的日志文件存储在同一存储桶中。当您启用某个特定分配的日志记录时，可为您的日志文件名称指定一个可选的前缀。

如果在给定时间内无用户访问您的内容，您在该时间内不会收到任何日志文件。

分析访问日志

由于单个媒体流的日志在多个文件中有记录，我们建议您将给定时段内接收的所有日志文件合并成一个文件。这样，您就可以更快更准确地分析该时段内的数据。

Important

我们建议您使用日志来了解内容请求的性质，而不是作为所有请求的完整描述。CloudFront 将尽力提供访问日志。特定请求的日志记录可能在实际处理请求后很长时间才予以提供，或者根本不提供。在极少数情况下，AWS 使用跟踪和计费系统中出现的使用情况可能不会出现在 CloudFront 访问日志中。

有关 CloudFront 访问日志（包括您可用于分析访问日志的工具建议）的更多信息，请参阅使用 CloudFront 日志记录 (p. 249)。

存储桶和文件所有权

您必须具有日志文件存储桶的 Amazon S3 FULL_CONTROL 权限。如果您是存储桶拥有者，默认情况下，您具有此权限。如果您不是，存储桶拥有者必须授予您 AWS 账户FULL_CONTROL权限。

您需要使用 CloudFront API 的 API 调用来启用日志记录。调用该 API 也将自动调用 Amazon S3 API，以更新存储桶的 ACL，以便得到 *AWS 数据源账户*的读写权限。此账户将日志文件写入存储桶。

每一个日志文件都有其自己的 ACL（不同于存储桶 ACL）。存储桶拥有者具有日志文件的 FULL_CONTROL 权限，分配所有者（如不是存储桶所有者）没有权限，而且数据源账户具有读写权限。

Note

撤销数据源账户的权限并不会禁用日志记录。如果您撤销这些权限但不禁用日志记录（您用 API 执行此操作），下一次当数据源账户需要向您的日志存储桶写入日志文件时我们会恢复这些权限。

如果您禁用日志记录，我们既不会撤销数据源账户对存储桶的读/写权限，也不会撤销其对日志文件的读/写权限。如果您希望撤销，可自己撤销。

如何更改日志记录设置

您可使用 CloudFront 控制台或使用 CloudFront API 启用或禁用日志记录，更改存储日志的 Amazon S3 存储桶以及更改日志文件的前缀：

- 有关使用 CloudFront 控制台更新 Web 或 RTMP 分配的信息，请参阅列出、查看及更新 CloudFront 分配 (p. 55)。
- 有关使用 CloudFront API 更新 Web 分配的信息，请转到 *Amazon CloudFront API 参考* 中的 PUT 分配配置。
- 有关使用 CloudFront API 更新 RTMP 分配的信息，请转到 *Amazon CloudFront API 参考* 中的 PUT 流分配配置。

您对日志记录设置的更改将在 12 小时内生效。

要使用 CloudFront API 更改

- Web 分配的访问日志设置，您必须使用 API 的 2009-04-02 版或更高版本。
- RTMP 分配的访问日志设置，您必须使用 API 的 2010-05-01 版或更高版本。
- Cookie 的访问日志设置，您必须使用 API 的 2012-07-01 版或更高版本。

Note

为了更方便在存储桶中列出密钥，Amazon S3 用户通常使用前缀后跟一个斜杆 (/) 作为分隔符。

如何删除 Amazon S3 存储桶中的日志文件

CloudFront 不会自动删除您在启用日志记录时指定的 Amazon S3 存储桶中的日志文件。有关删除 Amazon S3 存储桶中日志文件的信息，请参阅适用的 Amazon S3 文档：

- 使用 Amazon S3 控制台：请参阅 *Amazon Simple Storage Service 控制台用户指南* 中的删除对象。
- 使用 REST API：请参阅 *Amazon Simple Storage Service API 参考* 中的 DELETE 对象。
- 使用 SOAP API：请参阅 *Amazon Simple Storage Service API 参考* 中的 DeleteObject。

文件名格式和文件传送时机

文件名遵循此格式（使用 UTC 日期和小时）：

```
{bucket-name}.s3.amazonaws.com/{optional-prefix/}{distribution-ID}.{YYYY}-{MM}-{DD}-{HH}.{unique-ID}.gz
```

例如，如果您的存储桶名称为 `mylogs` 且前缀为 `myprefix/`，则您的文件名与以下内容相似：

```
mylogs.s3.amazonaws.com/myprefix/EMLARXS9EXAMPLE.2012-07-01-20.RT4KCN4SGK9.gz
```

如果您包含 `{optional-prefix/}` 的值，并且您的值不包含 `/`，CloudFront 将自动添加一个。如果您的值包含 `/`，CloudFront 将不会再另外添加一个。

CloudFront 使用 gzip 压缩每个日志文件，并将其保存在您指定的 Amazon S3 存储桶中。通常，CloudFront 会在收到相应请求后将日志文件保存 24 小时。根据提交请求的用户数量，CloudFront 可每小时保存一些日志文件。

Note

如果在给定时间内没有用户提交请求，您在该时间内不会收到任何日志文件。

日志文件格式

Topics

日志文件中的每个条目分别提供某个最终用户请求的详细信息。Web 和 RTMP 分配的日志文件是不相同的，但这两种类型的日志文件：

- 均使用 W3C 扩展日志文件格式。（有关更多信息，请访问 http://www.w3.org/TR/WD-logfile.html。）
- 包含制表符分隔的值。
- 包含不一定按时间顺序排列的记录。
- 包含两个标题行：一个具有文件格式版本，另一个列出了包含在每个记录中的 W3C 字段。
- 用 URL 编码的等效值替换字段值中的空格和非标准字符。

这些非标准字符由低于 32、高于 127 的 ASCII 码组成，再加上下表中的字符。URL 编码标准是 RFC 1738。有关更多信息，请访问 http://www.ietf.org/rfc/rfc1738.txt。

URL 编码值	字符
%3C	<
%3E	>
%22	"
%23	#
%25	%
%7B	{
%7D	}
%7C	|
%5C	\
%5E	^
%7E	~
%5B	[
%5D	]
%60	`
%27	'
%20	空格

Web 分配日志文件的格式

Web 分配的日志文件包括以下所列顺序的字段。

字段	描述
date	事件发生的日期 (UTC)，例如，2009-03-10。
time	服务器处理完请求的时间 (UTC)，例如，01:42:39。
x-edge-location	满足请求的节点。每个节点由三个字母的代码和任意分配的数字来标识，例如，DFW3。三个字母代码通常对应邻近节点的机场的国际航空协会机场代码。（这些缩写将来可能会更改。）关于节点的列表，请参阅 Amazon CloudFront 详细信息页面 http://aws.amazon.com/cloudfront。
sc-bytes	服务器到客户端的字节数，包括标头，例如，1045619。
c-ip	客户端 IP，例如，192.0.2.183。
cs-method	HTTP 访问方法。
cs(Host)	DNS 名称（请求中指定的 CloudFront 分配名称）。如果向别名记录 (CNAME) 发出请求，DNS 名称字段将包含基本的分配 DNS 名称，而非别名记录。

字段	描述
cs-uri-stem	URL 资源（例如，/images/daily-ad.jpg）。
sc-status	以下值之一： • HTTP 状态代码（例如，200）。有关更多信息，请参阅 CloudFront 缓存的 HTTP 4xx 和 5xx 状态码 (p. 89)。 • 000，表示在 CloudFront 响应请求之前查看者已关闭连接（例如，关闭浏览器标签）。
cs(Referer)	引用站点。
cs(User-Agent)	用户代理。
cs-uri-query	包含在连接字符串上的 URI 的查询字符串部分。当 URI 不包含查询字符串时，日志文件则在该请求的字段 cs-uri-query 中包含单个连字符 (-)。编码标准是 RFC 1738。有关更多信息，请参阅日志文件格式 (p. 148)。
cs(Cookie)	请求中的 Cookie 标头，包括名称值对和相关的属性。如果启用 Cookie 日志记录，无论您选择要将哪些 Cookie（无、全部或 Cookie 名称的白名单）转发到源，CloudFront 都将记录所有请求中的 Cookie。当请求中不包含 Cookie 标头时，日志文件则在该请求的字段 cs(Cookie) 中包含单个连字符 (-)。 有关 Cookie 的更多信息，请参阅 CloudFront 如何转发、缓存及记录 Cookie (p. 61)。
x-edge-result-type	请求的结果类型。结果类型包括： • Hit：CloudFront 从边缘缓存向查看者提供对象。 • RefreshHit：CloudFront 在边缘缓存中找到对象，但已过期，因此，CloudFront 联系源，以确认缓存具有该对象的最新版本。 • Miss：边缘缓存中的对象无法满足请求，因此 CloudFront 将请求转发给原始服务器并将结果返回给查看者。 • LimitExceeded：由于超出了 CloudFront 限制，请求被拒。 • CapacityExceeded：CloudFront 返回 503 错误，因为节点在收到请求时没有足够的容量来提供对象。 • Error：通常，这意味着请求导致客户端错误（sc-status 为 4*xx*）或服务器错误（sc-status 为 5*xx*）。 如果 sc-status 为 403，并且您将 CloudFront 配置为限制内容的地理分布，则请求可能来自某个受限制的位置。有关地理限制的更多信息，请参阅限制您的内容的地理分配 (p. 42)。 如果 sc-status 为 2*xx*，则客户端在下载完成前可能已断开。
x-edge-request-id	唯一地标识请求的加密字符串。

字段	描述
x-host-header	查看者在此请求的 Host 标头中包含的值。这是请求中的域名称： • 如果您使用 CloudFront 域名 (http://d111111abcdef8.cloudfront.net/logo.png)，`x-host-header` 列包含 CloudFront 指派给您的分配的域名，例如 `d111111abcdef8.cloudfront.net`。 • 如果使用备用域名 (http://example.com/logo.png)，`x-host-header` 列包含备用域名，例如 `example.com`。要使用备用域名，您必须将其添加到您的分配中。有关更多信息，请参阅 使用备用域名（别名记录）(p. 52)。
cs-protocol	查看者在请求中指定的协议：`http` 或 `https`。
cs-bytes	查看者在请求中包含的数据的字节数（客户端到服务器的字节数），包括标头。

Note

日志中不包含 URL 和查询字符串中的问号 (?)。

以下是 Web 分配的一个日志文件示例。

```
#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
07/01/2012 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 -
07/01/2012 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 -
```

RTMP 分配日志文件的格式

RTMP 访问日志中的每条记录代表一个回放事件，例如，连接、播放、暂停、停止、断开等等。因此，查看者每次观看视频时，CloudFront 都将产生多个日志记录。要关联源自同一流 ID 的日志记录，请使用 *`x-sid`* 字段。

Note

有些字段在所有事件中都存在，而另一些只出现在播放、停止、暂停、取消暂停及搜索事件中。当某个字段与给定的事件无关时，日志文件将包含单个连字符 (-)。

下表描述了 RTMP 分配日志文件中每条记录中出现的字段，不考虑事件的类型。字段以所列顺序出现在日志中。

字段	描述
date	事件发生的日期 (UTC)。

字段	描述
time	服务器接收到请求的时间 (UTC)，例如，01:42:39。
x-edge-location	回放事件发生的节点。每个节点由三个字母的代码和任意分配的数字来标识，例如，DFW3。三个字母代码通常对应邻近节点的机场的国际航空协会机场代码。（这些缩写将来可能会更改。）关于节点的列表，请参阅 Amazon CloudFront 详细信息页面 http://aws.amazon.com/cloudfront。
c-ip	客户端 IP，例如，192.0.2.183。
x-event	事件类型。这是连接、断开、播放、停止、暂停、取消暂停或搜索事件。
sc-bytes	截至事件发生时，从服务器发送到客户端的汇总字节数。
x-cf-status	指示事件状态的代码。目前，“OK”是此字段的唯一值。未来的新功能可能要求新的状态代码。
x-cf-client-id	可用于区分客户端的不透明的字符串标识符 此值对每个连接都是唯一的。
cs-uri-stem	URI 的主干部分，包括应用程序和应用程序实例。该部分有时称为 FMS *连接字符串*。例如，rtmp://shqshne4jdp4b6.cloudfront.net/cfx/st。
cs-uri-query	包含在连接字符串上的 URI 的查询字符串部分。
c-referrer	引用站点的 URI。
x-page-url	页面的 URL，SWF 与其链接在一起。
c-user-agent	用户代理。

以下字段只出现在播放、停止、暂停、取消暂停及搜索事件中。对于其他事件，这些字段将会包含单个破折号 (-)。这些字段在日志中出现在上面所列字段后面，并且以所列顺序出现。

字段	描述
x-sname	流名称。
x-sname-query	流查询字符串，如有。
x-file-ext	流类型，例如，FLV。
x-sid	流 ID。这是连接的唯一整数标识符。

Note

日志中不包含 URL 和查询字符串中的问号 (?)。

以下是 RTMP 分配的一个日志文件示例。

```
#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
-   -   -   -
```

访问日志的费用

访问日志记录 CloudFront 的一个可选功能。启用访问日志记录无额外费用。但是，在 Amazon S3 上存储和访问文件（您可随时删除他们）会产生常规的 Amazon S3 费用。有关 CloudFront 费用的更多信息，请参阅CloudFront 计费和使用 (p. 6)。

故障排除

Topics

我无法查看我的 Web 分配中的文件。

如果您无法查看您的 CloudFront Web 分配中的文件，下列主题介绍了一些常见的解决方案。

您是否同时注册了 CloudFront 和 Amazon S3？

要将 Amazon CloudFront 与 Amazon S3 源结合使用，您必须分别注册 CloudFront 和 Amazon S3。有关注册 CloudFront 和 Amazon S3 的更多信息，请参阅 CloudFront 入门 (p. 12)。

您的 Amazon S3 存储桶和对象权限是否设置正确？

如果您将 CloudFront 和 Amazon S3 源结合使用，则内容的原始版本存储在 Amazon S3 存储桶中。将 CloudFront 和 Amazon S3 结合使用的最简单方式是使 Amazon S3 中的所有对象公开可读。为此，您必须为您上传到 Amazon S3 中的每个对象明确启用公共读取权限。

如果您的内容不是公开可读的，您需要创建一个 CloudFront 原始访问标识，以便 CloudFront 可访问它。有关 CloudFront 原始访问标识的更多信息，请参阅使用原始访问标识限制访问您的 Amazon S3 内容 (p. 95)。

对象属性和存储桶属性是独立的。您必须明确授予对 Amazon S3 中每个对象的权限。没有从存储桶继承属性的对象及对象属性必须独立于存储桶进行设置。

您的备用域名（别名记录）配置是否正确？

如果您的域名已有对应的别名记录，请更新此记录或用指向分配域名的新记录替换它。

此外，还要确保您的别名记录指向您的分配的域名，而不是 Amazon S3 存储桶。您可确认您的 DNS 系统中的别名记录是否指向您的分配的域名。为此，请使用 DNS 工具，如 dig。（有关 dig 的信息，请访问 http://www.kloth.net/services/dig.php。）

以下显示了针对名为 images.example.com 的域名发出的示例 dig 请求以及响应中的相关部分。在 ANSWER SECTION 下，查看包含 CNAME 的行。如果别名记录右侧的值是您的 CloudFront 分配的域名，则说明您域名的别名记录设置是正确的。如果是您的 Amazon S3 原始服务器存储桶或其他某个域名，则说明别名记录设置不正确。

```
[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.
...
...
```

有关别名记录的更多信息，请参阅使用备用域名（别名记录）(p. 52)。

您引用的是 CloudFront 分配的正确 URL 吗？

请确保您引用的 URL 使用您的 CloudFront 分配域名（或您的别名记录），而不是您的 Amazon S3 存储桶或自定义源。

您需要帮助排查自定义源方面的问题吗？

如果您需要 AWS 帮助您排查自定义源方面的问题，我们将可能需要检查您请求中的 X-Amz-Cf-Id 标头条目。如果您尚未记录这些条目，您将来可能需要考虑记录它们。有关更多信息，请参阅 使用 Amazon EC2 和其他自定义源的要求和建议 (p. 41)。

我无法查看我的 RTMP 分配中的文件。

如果您无法查看 RTMP 分配中的文件，那么请问您的 URL 和您的回放客户端配置是否正确？RTMP 分配要求您使用 RTMP 协议而不是 HTTP，而且您必须对您的回放客户端作出一些微小的配置更改。有关创建 RTMP 分配的信息，请参阅创建 RTMP 分配 (p. 22)。

错误消息："Certificate: <certificate-id> is being used by CloudFront (证书: <证书 ID> 正在被 CloudFront 使用)"。

问题：您在尝试从 IAM 证书存储区中删除 SSL 证书，此时您收到消息称"Certificate: <certificate-id> is being used by CloudFront (证书: <证书 ID> 正在被 CloudFront 使用)"。

解决方案：每个 CloudFront Web 分配均必须与默认 CloudFront 证书或自定义 SSL 证书关联。在删除 SSL 证书之前，您需要轮换 SSL 证书（将当前的自定义 SSL 证书替换为其他自定义 SSL 证书），或从使用自定义 SSL 证书恢复为使用默认的 CloudFront 证书。执行如下相关章节中的步骤：

- 轮换 SSL 证书 (p. 139)
- 从自定义 SSL 证书恢复为默认 CloudFront 证书 (p. 140)

对 CloudFront 进行负载测试

传统的负载测试方法对 CloudFront 并不十分适用，因为 CloudFront 使用 DNS 在分散于不同地理位置的节点间以及每个节点内部平衡负载。当客户端向 CloudFront 请求内容时，客户端会收到包含一组 IP 地址的 DNS 响应。如果您采用的测试方法是只向 DNS 返回的其中一个 IP 地址发送请求，那么您测试的仅仅是一个 CloudFront 节点中的一小部分资源，这并不能准确体现实际的流量规律。根据所请求的数据量，以这种方式进行测试可能会造成这一小部分 CloudFront 服务器超载且性能下降。

CloudFront 旨在通过扩展来满足在多个地理区域具有不同客户端 IP 地址和不同 DNS 解析程序的查看器之需。要执行能准确评估 CloudFront 性能的负载测试，我们建议您执行以下所有操作：

- 从多个地理区域发送客户端请求。
- 配置您的测试，以使每个客户端发出独立的 DNS 请求；这样每个客户端就会从 DNS 分别收到一组不同的 IP 地址。
- 对于每个发出请求的客户端，使用 DNS 返回的该组 IP 地址分散发出您的客户端请求，从而确保在 CloudFront 节点中的多台服务器间分配负载。

发出 API 请求

Topics

本节介绍如何向您用来创建和管理分配的 CloudFront API 发出 REST 请求。各个主题分别为您介绍了请求的组成部分、响应的内容以及如何对请求进行身份验证。

终端节点

与其他 Amazon 服务不同的是，CloudFront 在该服务的运行区域（例如新加坡、欧洲/都柏林、美国/东部等）没有多个终端节点。这是因为 CloudFront 分配不是像 Amazon S3 存储桶和 Amazon EC2 实例那样的区域资源。相反，CloudFront 可从其众多节点之一提供内容。这意味着 CloudFront 分配具有单个终端节点：具体分配的原始服务器位置。

因此，当您发出 REST 请求时，您需采用以下格式，其中 *<distribution>* 是您在请求中要求采取行动处理的分配。

```
cloudfront.amazonaws.com/2013-11-11/<distribution>
```

相关主题

- REST 请求 (p. 159)
- Amazon CloudFront 网络（AWS 网站上所有 Amazon CloudFront 节点的列表）
- 区域和终端节点（Amazon Web Services 一般参考中有关 AWS 产品终端节点和地区的信息）

AWS 对编程语言的支持

对于那些喜欢使用特定语言的 API 而不愿意使用 CloudFront 的 REST API 构建应用程序的软件开发人员，AWS 为他们提供了库、示例代码、教程和其他资源。这些库文件提供基本功能（不包括在 CloudFront 的 REST API 中），例如请求身份验证、请求重试和错误处理，让您可以更轻松地开始使用。现已推出适用以下语言的库和资源：

- Java
- PHP
- Python
- Ruby
- Windows 和 .NET

有关以所有语言推出的库和示例代码，请转到示例代码和库。

REST 请求

按照 RFC 2616 的定义，Amazon CloudFront REST 请求属于 HTTPS 请求。有关更多信息，请访问 http://www.ietf.org/rfc/rfc2616.txt。本节介绍 CloudFront REST 请求的结构。有关您可以执行的操作的详细说明，请访问 Amazon CloudFront API 参考。

典型的 REST 操作包括向 CloudFront 发送单个 HTTPS 请求，然后等待 HTTP 响应。像任何 HTTP 请求一样，发往 CloudFront 的 REST 请求也包含请求方法、URI、请求标头，有时还包含查询字符串或请求正文。响应包含 HTTP 状态码、响应标头，有时还包含响应正文。

请求 URI

请求 URI 始终以正斜杠开头，然后是您使用的 CloudFront API 版本，例如，2013-11-11。URI 的其余部分表示您希望采取措施处理的特定资源。例如，以下是您在创建新分配时使用的 URI：

```
/2013-11-11/distribution
```

有关使用 CloudFront API 创建分配的更多信息，请访问 *Amazon CloudFront API 参考*中的 POST 分配。

请求标题

下表列出了 CloudFront REST 请求使用的 HTTP 标头。

标题名称	描述	必需
`Authorization`	对请求进行身份验证时所需要的信息。有关更多信息，请参阅 对 REST 请求进行身份验证 (p. 164)。	是
`Content-Length`	根据 RFC 2616 标准计算得出的消息长度（不计标头）。 条件：如果请求正文本身包含信息，则为必需项（大多数工具包都会自动添加此标头）。	条件
`Content-Type`	资源的内容类型。示例：`text/plain`。 条件：对于 POST 和 PUT 请求，则为必需项。	条件

标题名称	描述	必需
Date	用于创建 Authorization 标头中所包含的签名的日期。格式必须是 RFC 2616 第 3.1.1 节中规定的完整日期格式之一，例如，Wed, 05 Apr 2006 21:12:00 GMT。有关更多信息，请访问 RFC 2616 规范。 条件：除非您提供 x-amz-date 标头，否则为必需项。有关请求时间戳的更多信息，请参阅请求时间戳 (p. 160)。	条件
Host	所请求的主机。该值必须为 cloudfront.amazonaws.com。 条件：对于 HTTP 1.1，则为必需项。大多数工具包都会自动添加此标头。	条件
x-amz-date	用于创建 Authorization 标头中所包含的签名的日期。格式必须是 RFC 2616 第 3.1.1 节中规定的完整日期格式之一，例如，Wed, 05 Apr 2006 21:12:00 GMT。有关更多信息，请访问 RFC 2616 规范。 条件：如果您不提供 Date 标头，则为必需项。有关更多信息，请参阅 请求时间戳 (p. 160)。	条件

请求时间戳

您必须在 HTTPDate 标头或 AWS x-amz-date 标头中提供时间戳（有些 HTTP 客户端库不允许您设置 Date 标头）。当存在 x-amz-date 标头时，系统会在对请求进行身份验证时忽略任何 Date 标头。

时间戳必须在收到请求时 AWS 系统时间之后的 15 分钟内。如果不在此时间范围内，请求将失败，并出现 RequestExpired 错误代码。这是为了防止对方重放您的请求。

请求主体

很多 CloudFront API 操作都要求您将 XML 包括在请求正文内。XML 符合 CloudFront 架构。本指南中介绍 API 操作的主题说明了请求中所需的 XML 结构。

示例请求

下面的示例请求在 CloudFront 系统中创建一项分配。

```
POST /2013-11-11/distribution HTTP/1.1
Host: cloudfront.amazonaws.com
Authorization: [AWS authentication string]
Date: Thu, 17 May 2012 19:37:58 GMT
[Other required headers]

<?xml version="1.0" encoding="UTF-8"?>
<DistributionConfig xmlns="http://cloudfront.amazonaws.com/doc/2013-11-11/">
   <CallerReference>example.com2012-04-11-5:09pm</CallerReference>
   <Aliases>
      ...
   </Aliases>
   <DefaultRootObject>index.html</DefaultRootObject>
   <Origins>
      ...
   </Origins>
```

```
   <CacheBehaviors>
      ...
   </CacheBehaviors>
   <Comment>example comment</Comment>
   <Logging>
      ...
   </Logging>
   <PriceClass>PriceClass_All</PriceClass>
   <Enabled>true</Enabled>
</DistributionConfig>
```

相关主题

- 对 REST 请求进行身份验证 (p. 164)
- REST 响应 (p. 162)

REST 响应

Amazon CloudFront 响应就是标准的 HTTP 响应。部分 CloudFront 操作以 HTTP 标头形式或响应正文中的 XML 形式返回 CloudFront 所特有的特殊信息。具体细节在与特定操作对应的 API 参考主题中进行了阐述。

请求 ID

Each response contains a request ID that you can use if you need to troubleshoot a request with AWS. The ID is contained in an HTTP header called `x-amz-request-id`. An example of a request ID is `647cd254-e0d1-44a9-af61-1d6d86ea6b77`.

响应示例

以下示例显示了创建分配时的响应。

```
201 Created
Location: https://cloudfront.amazonaws.com/2013-11-11/distribution/EDFDVBD6EX
AMPLE
x-amz-request-id: request_id


<?xml version="1.0" encoding="UTF-8"?>
<Distribution xmlns="http://cloudfront.amazonaws.com/doc/2013-11-11/">
   <Id>EDFDVBD6EXAMPLE</Id>
   <Status>InProgress</Status>
   <LastModifiedTime>2012-05-19T19:37:58Z</LastModifiedTime>
   <DomainName>d111111abcdef8.cloudfront.net</DomainName>
   <DistributionConfig>
      <CallerReference>example.com2012-04-11-5:09pm</CallerReference>
      <Aliases>
         ...
      </Aliases>
      <DefaultRootObject>index.html</DefaultRootObject>
      <Origins>
         ...
      </Origins>
      <CacheBehaviors>
         ...
      </CacheBehaviors>
      <Comment>example comment</Comment>
      <Logging>
         ...
      </Logging>
      <PriceClass>PriceClass_All</PriceClass>
      <Enabled>true</Enabled>
   </DistributionConfig>
</Distribution>
```

错误响应

如果 REST 请求导致错误，则 HTTP 回复具有：

- XML 错误文档作为响应主体

- Content-Type 标头：application/xml
- 适当的 3xx、4xx 或 5xx HTTP 状态码

以下是 REST 错误响应中的 XML 错误文档的示例。

```
<ErrorResponse xmlns="http://cloudfront.amazonaws.com/doc/2013-11-11/">
   <Error>
      <Type>Sender</Type>
      <Code>InvalidURI</Code>
      <Message>Could not parse the specified URI.</Message>
   </Error>
   <RequestId>410c2a4b-e435-49c9-8382-3770d80d7d4c</RequestId>
</ErrorResponse>
```

相关主题

- 错误（在 *Amazon CloudFront API 参考*中）
- REST 请求 (p. 159)
- 对 REST 请求进行身份验证 (p. 164)

对 REST 请求进行身份验证

CloudFront 要求通过对请求进行签名，验证所发送的每个请求的身份。要对请求进行签名，您需要使用加密哈希函数计算出数字签名，此函数可根据输入返回一个哈希值。输入内容包括您的请求文本和私有访问密钥。哈希函数返回哈希值，您将该值包含在请求中，作为签名。签名是请求的 `Authorization` 标头的一部分。

收到您的请求后，CloudFront 使用与您用于对该请求进行签名的相同哈希函数和输入重新计算签名。如果所得签名与该请求中的签名相匹配，则 CloudFront 处理该请求。否则，请求将被拒绝。

CloudFront API 的 2013-05-12 版本及更高版本要求您使用 AWS 签名版本 4 对请求进行身份验证。

Important

如果您使用的是 CloudFront API 的 2012-07-01 版本或更低版本，则您必须使用更低版本的 AWS 签名来对请求进行身份验证。有关更多信息，请参阅 CloudFront 文档存档中适用版本的*Amazon CloudFront 开发人员指南*中的“对 REST 请求进行身份验证”。

计算签名的过程可分为三个任务：

- 任务 1：创建规范请求

  按照*Amazon Web Services General Reference*中的任务 1：针对签名版本 4 创建规范请求中所述，以规范格式创建 HTTP 请求。
- 任务 2：创建待签字符串

  创建一个字符串，将该字符串用作您的加密哈希函数输入值中的一项。该字符串称为*待签字符串*，由哈希算法名称、请求日期、*证书范围*字符串以及来自上一任务的规范化请求组合而成。“证书范围”字符串本身由日期、地区和服务信息组合而成。

  对于 `Credential` 参数，请指定：
  - 您要将请求发送到的终端节点的代码，即 `us-east-1`。
  - `cloudfront`（表示服务缩写）

  例如：

  ```
  Credential=AKIAIOSFODNN7EXAMPLE/20130605/us-east-1/cloudfront/aws4_request
  ```
- 任务 3：创建签名

  使用加密哈希函数为您的请求创建签名，该函数接受两种输入字符串：*待签字符串*和*派生密钥*。*派生密钥*的计算方法是，以您的私有访问密钥开头并使用*证书范围*字符串来创建一系列基于哈希的消息身份验证代码 (HMAC)。

CloudFront 教程

下面的教程介绍了如何将 CloudFront 用于实时流、geoblocking 和 RTMP 流。

实时流

Geoblocking

RTMP 流

使用 CloudFront 和 Adobe Media Server 5.0 的实时 HTTP 流

Topics

借助 Amazon Web Services 实时流，您可以使用 Adobe Media Server 5.0 版流式传输现场表演、网络研讨会和其他活动。本教程将引导您完成配置实时流和 Adobe Media Server 5.0 的过程。

概述

Adobe Media Server 5.0 支持两种 HTTP 流格式：

- HLS（HTTP 实时流），受 iOS 设备支持
- HDS（HTTP 动态流），受 Flash 应用程序支持

以下说明了 Adobe Media Server 如何与 CloudFront 协同运行，以便实时流式传输某一事件：

1. 如本教程中所述，您需要使用 AWS CloudFormation 配置一个运行 Adobe Media Server 5.0 的 Amazon EC2 实例，并创建一个 CloudFront 分配。
2. 您需要使用数字摄像机捕获您的事件，例如，使用笔记本电脑上的摄像头。
3. 您应在事件发生地使用编码器（例如，Adobe Flash Media Live Encoder）压缩原始视频源，并将其发送至 Adobe Media Server。（Flash Media Live Encoder 提供免费下载，可用于 Windows 和 Mac OS。）
4. Adobe Media Server 会将视频流分割成一系列较小的文件。此服务器便是您的 CloudFront 分配的源。
5. 当您的用户浏览到您为他们提供的用来查看事件的 CloudFront URL 后，CloudFront 会将其路由到最近的节点（根据延迟）。
6. 此节点将从 Adobe Media Server 请求视频流。
7. Adobe Media Server 以小文件的形式向 CloudFront 节点返回视频流。
8. CloudFront 节点向发出请求的用户供应视频流，并缓存这些小文件，以便在后续请求实时流时加快响应速度。

本教程概述了如何将 CloudFront 与 Amazon EC2 实例上运行的 Adobe Media Server 相集成。有关 Adobe Media Server 以及用于实时流的 AWS 服务的更多信息，请参阅以下内容：

- 有关本教程中未包含的 Adobe Media Server 选项的更多信息，请参阅其他文档 (p. 182)。
- 有关可用 Adobe Media Server 功能的信息，请参阅 Amazon Web Services 上的 Adobe Media Server 5。
- 要查看 Adobe Media Server 5.0 中的新功能，请参阅 Adobe 网站上的 Adobe Media Server 5.0.1 新增功能。
- 有关如何管理和保护 Amazon EC2 实例的更多信息，请参阅 Amazon EC2 文档。
- 有关 AWS CloudFormation 的更多信息，请参阅 AWS CloudFormation 文档。
- 有关更多帮助，请参阅 常见问题 (p. 176)。

配置实时流的步骤

要设置 Amazon Web Services (AWS) 实时流，请检查 Adobe Flash Player 的系统要求。然后执行以下几个部分中的步骤：

1. 创建 Amazon Web Services 账户 (p. 167)
2. 创建 Amazon EC2 密钥对 (p. 167)
3. 订阅 Adobe Media Server (p. 168)
4. 创建 AWS CloudFormation 实时流堆栈 (p. 168)
5. 验证 Adobe Media Server 是否正在运行 (p. 170)
6. 设置 Adobe Flash Media Live Encoder 以发布实时流 (p. 170)
7. 在 Web 应用程序中为 Amazon CloudFront 实时 HTTP 流嵌入 Flash 媒体播放 (p. 174)
8. 删除 AWS CloudFormation 实时流堆栈 (p. 176)

创建 Amazon Web Services 账户

如果您已有 AWS 账户，请跳至创建 Amazon EC2 密钥对 (p. 167)。如果您没有 AWS 账户，请通过以下步骤创建一个。

Note

AWS 会在您创建账户时自动为该帐户注册所有服务。您只需为使用的服务付费。

创建 AWS 账户

1. 转至 http://aws.amazon.com，然后单击 Create an AWS Account (创建 AWS 账户)。
2. 按照屏幕上的说明进行操作。

   作为注册流程的一部分，您会收到一个电话，需要您使用电话键盘输入一个 PIN 码。

下一步：创建 Amazon EC2 密钥对 (p. 167)

创建 Amazon EC2 密钥对

如果您在要配置实时流的 Amazon EC2 区域中已有 Amazon EC2 密钥对，请跳至订阅 Adobe Media Server (p. 168)。如果您在该区域无密钥对，请执行以下步骤。

密钥对是一种类似于密码的安全证书。在本过程的后面部分，您将在创建 AWS CloudFormation 实时流堆栈时指定一个密钥对。在配置实时流后，您可使用密钥对安全地连接到 Amazon EC2 实例。

创建 Amazon EC2 密钥对

1. 登录 AWS 管理控制台，并通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在 Region (区域) 列表中，单击要在其中创建密钥对的区域。

   您要创建密钥对的区域必须与要在本过程后面部分中创建 AWS CloudFormation 实时流堆栈的区域相同。我们建议，在创建密钥对和实时流堆栈时，应选择距离将要执行流式传输的用户最近的区域。
3. 在“Navigation (导航)”窗格中，单击 Key Pairs (密钥对)。
4. 在 Key Pairs (密钥对) 窗格中，单击 Create Key Pair (创建密钥对)。

5. 在“Create Key Pair (创建密钥对)”对话框中，输入密钥对的名称并记录下该名称。在实时流设置过程的后面部分中，您将在创建 AWS CloudFormation 实时流堆栈时输入此值。
6. 单击 Create (创建)。
7. 在“Opening <key_pair_name>.pem (正在打开 <key_pair_name>.pem)”对话框中，将 .pem 文件保存到计算机上的安全位置。

   Important

   您将只有这一次机会来下载和保存您的私有密钥。

8. 单击 Close (关闭) 以关闭“Create Key Pair (创建密钥对)”对话框。

下一步：订阅 Adobe Media Server (p. 168)

订阅 Adobe Media Server

执行以下步骤，以便使用您的 AWS 账户为 Amazon Web Services 订阅 Adobe Media Server。

每月订阅费为 5.00 USD。支付此费用后，您可以运行无限数量的 Adobe Media Server 实例。除每月订阅费之外，还有每小时使用费用和数据传输费用。在执行以下步骤时，您可以查看详细的价目表。

为 Amazon Web Services 订购 Adobe Media Server

1. 转至 Amazon Web Services 上的 Adobe Media Server 5 页面。
2. 单击 Subscribe Now (立即订阅)。
3. 按照屏幕上的说明进行操作。
4. 阅读定价条款，然后单击页面底部的 Place Your Order (下单)。

下一步：创建 AWS CloudFormation 实时流堆栈 (p. 168)

创建 AWS CloudFormation 实时流堆栈

下面的步骤将使用 AWS CloudFormation 模板来创建一个堆栈，用于启动实时流所需的 AWS 资源，包括一个 Amazon EC2 实例和一个 CloudFront 分配。

Important

当您创建用于部署 Amazon EC2 实例的 AWS CloudFormation 堆栈时，将会开始产生按小时计算的 Amazon EC2 实例费用。无论您是否使用 Amazon EC2 实例来流式传输实时视频，费用都会一直累积，直到您删除 AWS CloudFormation 堆栈为止。有关更多信息，请参阅 Amazon Web Services 上的 Adobe Media Server 定价。当您的实时事件结束时，请删除为实时流而创建的堆栈。这将删除为您的实时流事件而创建的 AWS 资源，并停止 AWS 资源计费。有关更多信息，请参阅 删除 AWS CloudFormation 实时流堆栈 (p. 176)。

有关 AWS CloudFormation 的更多信息，请参阅 AWS CloudFormation 文档。

创建 AWS CloudFormation 实时流堆栈

1. 要启动“Create Stack (创建堆栈)”向导，请单击以下 Amazon EC2 区域之一：

   - 在美国东部（弗吉尼亚）创建堆栈
   - 在美国西部（俄勒冈）创建堆栈
   - 在美国西部（加利福尼亚北部）创建堆栈

- 在欧洲（爱尔兰）创建堆栈
- 在亚太地区（新加坡）创建堆栈
- 在亚太地区（东京）创建堆栈
- 在亚太地区（澳大利亚）创建堆栈
- 在南美洲（圣保罗）创建堆栈

此向导将会启动，并且 Provide a Template URL (提供模板 URL) 字段中会自动显示相应的 URL。

Note

如果希望用户使用在您自己的域上托管的基于 Flash 的播放器来查看您的实时流，请参阅 如何为在我自己的域上托管的 Flash 流更新 crossdomain.xml？ (p. 178)。

2. 如果您尚未登录 AWS Management Console，请在出现提示时登录。
3. *可选：*更改 Stack Name (堆栈名称)。堆栈名称不得包含空格，必须是 AWS 账户内的唯一名称。
4. 请勿更改 Template (模板) 选项或 Provide a Template URL (提供模板 URL) 内的地址。
5. *可选：*要配置 SNS 通知、要指定您愿意等待的堆栈创建时间以及要选择是否在堆栈创建失败后回滚更改，请选中 Show Advanced Options (显示高级选项)，然后根据需要调整设置。
6. 单击 Continue (继续)。
7. 在“Specify Parameters (指定参数)”页面上，对于 KeyPair (密钥对)，请输入您在步骤 1 中选择的区域中的一个 Amazon EC2 密钥对名称。该密钥对必须与您当前登录的账户相关联。如果您在执行创建 Amazon EC2 密钥对 (p. 167)中的步骤时创建了密钥对，请输入该密钥对的名称。
8. 在 AMSAdminPassword 中，为您在步骤 9 中指定的用户名输入密码（最少 8 个字符）。
9. 对于 StreamName，请为您的实时流输入一个短名（无空格）。
10. 对于 AMSAdminUserName 字段，请输入一个用户名。在您的 Amazon EC2 Adobe Media Server 实例完成创建后，您将使用此值登录 Adobe 媒体管理控制台。
11. 对于 InstanceType，请输入一种实例类型。默认值为 *m1.xlarge*。

    实例类型决定着 Adobe Media Server 实例的定价。下表列出了实时流所支持的 Amazon EC2 实例类型。

    有关 Amazon EC2 实例类型的更多信息，请参阅 *Amazon Elastic Compute Cloud 用户指南* 中的可用实例类型。

    有关定价信息，请转至 Amazon Web Services 上的 Adobe Media Server 5 页面，然后单击 North America (北美洲) 或 International (国际) 选项（如果适用）。

Amazon EC2 和 Adobe Media Server 实例类型	InstanceType 代码
大型	m1.large
超大型	m1.xlarge
内存增强型超大型	m2.xlarge
CPU 增强型超大型	c1.xlarge
内存增强型双倍超大型	m2.2xlarge
内存增强型四倍超大型	m2.4xlarge

12. 单击 Continue (继续)。
13. *可选：*在“Add Tags (添加标签)”页面上，添加一个或多个标签，然后单击 Continue (继续)。

14. 检查堆栈的设置。如果您对设置感到满意，请单击 Continue (继续)，然后 AWS CloudFormation 将创建堆栈。

    堆栈创建可能需要花费几分钟。要跟踪堆栈的创建进度，请选择堆栈，然后单击底部框架中的 Events (事件) 选项卡。如果 AWS CloudFormation 无法创建堆栈，则 事件 选项卡将列出错误消息。

    当您的堆栈准备就绪后，在顶部框架中，堆栈的状态将变为 CREATE_COMPLETE。

    当您的堆栈完成创建后，请单击 Outputs (输出) 选项卡，然后将显示堆栈创建输出结果。当您本过程后面部分中设置 Adobe Flash Media Live Encoder 时，您将用到这些值。

下一步：验证 Adobe Media Server 是否正在运行 (p. 170)

验证 Adobe Media Server 是否正在运行

在 AWS CloudFormation 创建堆栈后，请执行以下步骤来验证 Adobe Media Server 是否正在您通过 AWS CloudFormation 而配置的 Amazon Amazon EC2 实例上运行。

验证 Adobe Media Server 是否正在运行

1. 通过以下网址打开 AWS CloudFormation 控制台：https://console.aws.amazon.com/cloudformation/。
2. 在顶部窗格中，选择您在创建 AWS CloudFormation 实时流堆栈 (p. 168)中创建的堆栈。
3. 在底部窗格中，单击 Outputs (输出) 选项卡。
4. 单击 AMSServer 密钥的值，这是您在执行 创建 AWS CloudFormation 实时流堆栈 (p. 168) 中的步骤时所配置的 Amazon EC2 实例的 URL。
5. 此时将显示 Adobe Media Server 页面并开始流式传输内容，这表示 Adobe Media Server 正在运行。

   如果未开始流式传输，请返回概述 (p. 166)，并验证您在前四项任务中指定的值是否正确。

   如果值全部正确，但仍未开始流式传输，请参阅 当流无法启动时，我如何排查 Amazon EC2 实例的问题？ (p. 181)。

下一步：设置 Adobe Flash Media Live Encoder 以发布实时流 (p. 170)

设置 Adobe Flash Media Live Encoder 以发布实时流

Amazon Web Services 上的 Adobe Media Server 包括一个名为 livepkgr 的应用程序，它可以将所发布的流打包，以便使用 HTTP 动态流 (HDS) 和 HTTP 实时流 (HLS) 进行传送。

以下步骤展示了如何设置 Adobe Flash Media Live Encoder (FMLE)，以便将实时流发布到 Adobe Media Server 5.0 上的 livepkgr 应用程序。

Note

Flash Media Live Encoder 的 Windows 版本不支持 AAC 音频格式。要添加 AAC 支持，Adobe 建议您购买 MainConcept AAC 编码器。

在 Flash Media Live Encoder 中指定实时流设置

1. 登录到您将用来广播实时流的计算机。
2. 打开 Web 浏览器，然后转至 Adobe Flash Media Live Encoder 页面。
3. 下载并安装 Flash Media Live Encoder。

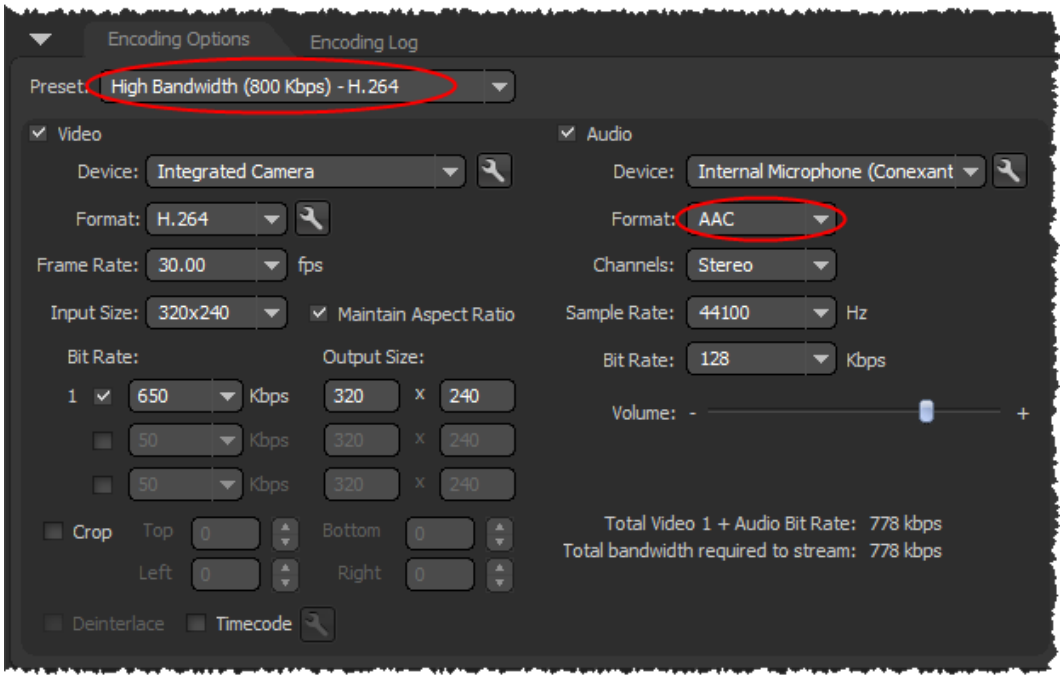
> **Note**
>
> Flash Media Live Encoder 是免费的，但您需要 Adobe 账户（同样免费）才能下载。

4. 在文本编辑器中打开 Flash Media Live Encoder `config.xml` 文件。默认安装位置取决您的操作系统：
   - 32 位 Windows：`C:\Program Files\Adobe\Flash Media Live Encoder 3.2`。
   - 64 位 Windows：`C:\Program Files (x86)\Adobe\Flash Media Live Encoder 3.2\Conf`。
   - Macintosh：`Applications:Adobe:Flash Media Live Encoder 3.2`。

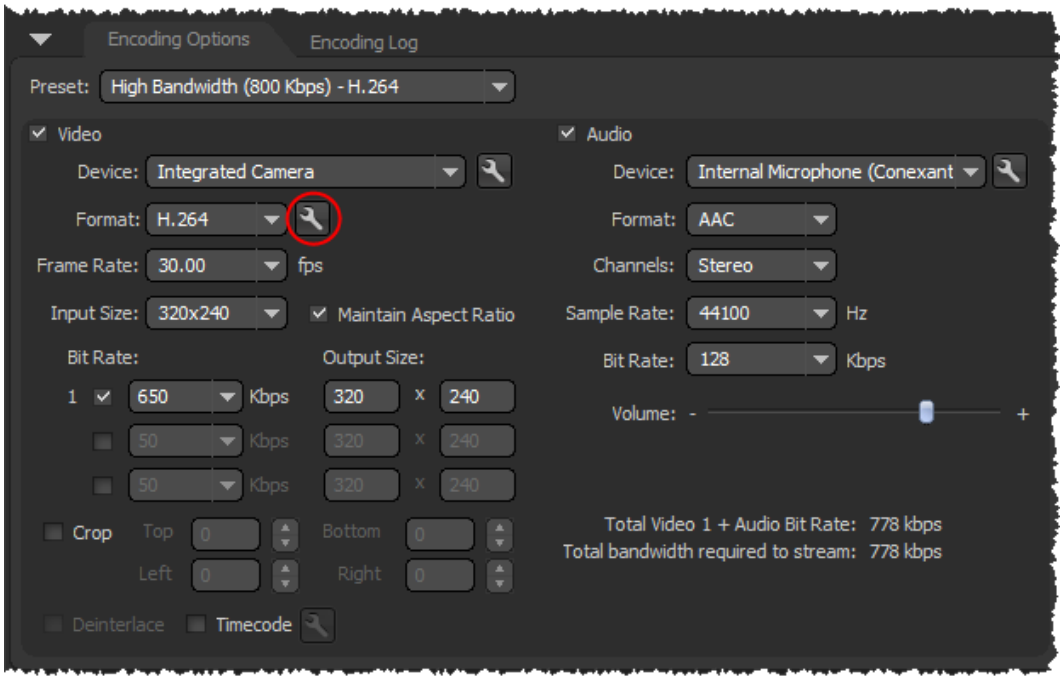
5. 在 `config.xml` 中，将以下 `<enable>` 元素的值设置为 `true`：

```
<flashmedialiveencoder_config>
   ...
   <mbrconfig>
      ...
      <streamsynchronization>
         ...
         <!-- "true" to enable this feature, "false" to disable. -->
         <enable>true</enable>
```

6. 保存文件。
7. 运行 Flash Media Live Encoder。
8. 在 Encoding Options (编码选项) 选项卡上，对于 Preset (预设)，请选择 High Bandwidth (800 Kbps) - H.264 (高带宽 (800 Kbps) - H.264)。
9. 在 Encoding Options (编码选项) 选项卡上，在 Audio (音频) 复选框下，为 Format (格式) 选择 AAC。

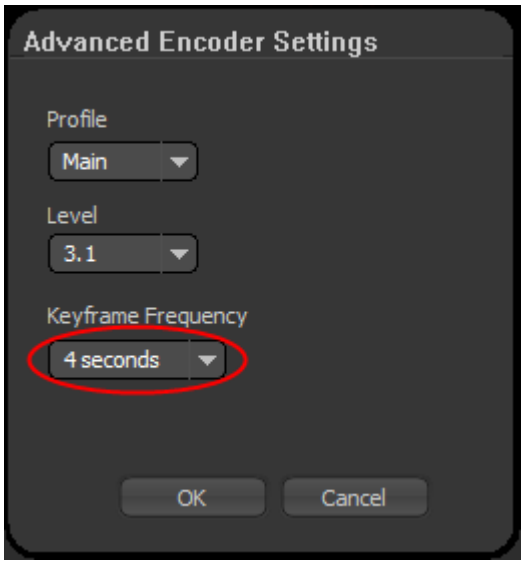


10. 在 Encoding Options (编码选项) 选项卡的 Video (视频) 部分中，单击 Format (格式) 列表右侧的扳手图标，以便打开 Advanced Encoder Settings (高级编码器设置) 对话框。



11. 在 Advanced Encoder Settings (高级编码器设置) 对话框中，对于 Keyframe Frequency (关键帧频率)，请选择 4 Seconds (4 秒)。

    您也可以使用 `applications/livepkgr/events/_definst_/liveevent/Event.xml` 文件中 <FragmentDuration> 元素值的倍数。默认的 <FragmentDuration> 值为 4000 毫秒（4 秒）。



12. 单击 OK (确定) 以保存设置并返回到主页。Preset (预设) 列表的选择将更改为 Custom (自定义)。
13. 通过以下网址打开 AWS CloudFormation 控制台：https://console.aws.amazon.com/cloudformation/。
14. 选中您为实时流而创建的堆栈的复选框。
15. 在底部窗格中，单击 Outputs (输出) 选项卡。

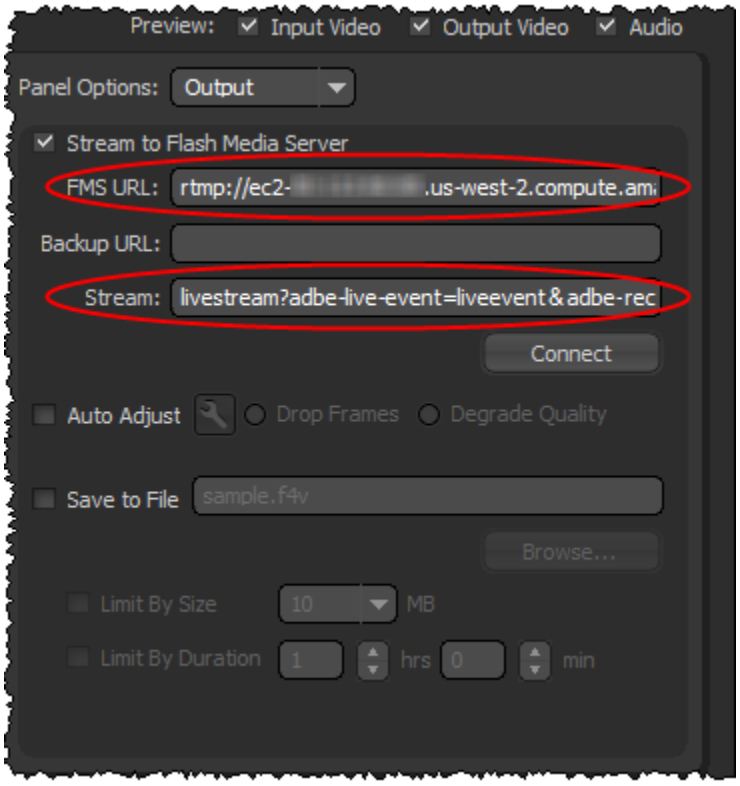
16. 复制 AMSURL 密钥的值，例如，*rtmp://ec2-00-11-22-33.us-west-1.compute.amazonaws.com/livepkgr*。
17. 在 Flash Media Live Encoder 的“Stream to Flash Media Server (流式传输到 Flash Media Server)”部分中，在 FMS URL 设置中，粘贴您从 AWS CloudFormation 控制台复制的 AMSURL 密钥的值。
18. 在 AWS CloudFormation 控制台中，复制 Stream (流) 密钥的值，例如，*livestream?adbe-live-event=liveevent*。
19. 在 Flash Media Live Encoder 中的 Stream (流) 设置中，粘贴您从 AWS CloudFormation 控制台复制的 Stream (流) 密钥的值。

    Note

    如果您预计必须停止和重启实时流，请在 Stream (流) 字段中输入以下值：

    `livestream?adbe-live-event=liveevent&adbe-record-mode=record`

    如果您以记录模式 (`adbe-record-mode=record`) 发布实时流，随后停止和重启该流，则 Adobe Media Server 将删除先前的流并启动新流，而不是在重启时附加到先前的流。但是，如果您没有使用记录模式并停止实时流，则必须重新配置实时流，然后才能重启该流。



20. 取消选中 Save to File (保存到文件)。
21. 单击 Connect (连接) 以连接到您的 Adobe Media Server 实例。
22. 单击 Start (开始)，以便开始对您的实时流进行编码，并将其发布到 Adobe Media Server 实例上的 livepkgr 应用程序。

下一步：在 Web 应用程序中为 Amazon CloudFront 实时 HTTP 流嵌入 Flash 媒体播放 (p. 174)

在 Web 应用程序中为 Amazon CloudFront 实时 HTTP 流嵌入 Flash 媒体播放

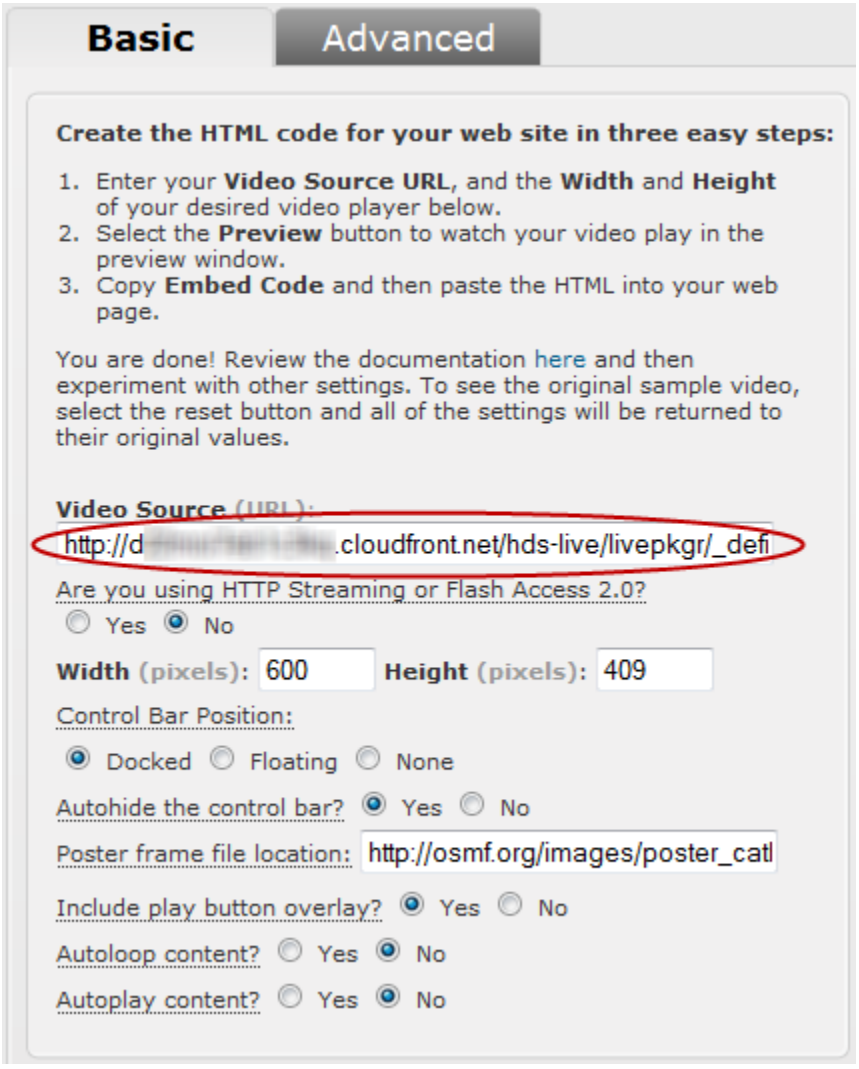
执行相应的步骤，以便获得要在您的 Web 页面中为实时流而包含的嵌入代码：

通过 CloudFront 为您的 HTTP 流嵌入 Flash 媒体播放

1. 通过以下网址打开 AWS CloudFormation 控制台：https://console.aws.amazon.com/cloudformation/。
2. 选择实时流堆栈。
3. 在底部窗格中，单击 Outputs (输出) 选项卡。
4. 复制 LiveHDSManifest 密钥的值，例如，*http://d123.cloudfront.net/hds-live/livepkgr/_definst_/liveevent/livestream.f4m*。
5. 单击 FlashMediaPlayback 密钥的值，以打开 Flash 媒体播放设置网页。
6. 在 Flash 媒体播放设置页面上，对于 Video Source (视频源)，请粘贴您在步骤 4 中从 AWS Management Console 复制的值。

Note

Flash 媒体播放需要安装 Flash Player 10.1 以支持 HTTP 动态流。



7. 对于 Are you using HTTP Streaming or Flash Access 2.0 (您是否正在使用 HTTP 流或 Flash Access 2.0)，请单击 Yes (是)。
8. 删除 Poster frame file location (标识帧文件位置) 的值（如果有）。
9. 单击 Preview (预览)，以便更新 Preview Embed Code (预览嵌入代码) 的值。

10. 播放视频，确保您满意当前的设置。
11. 根据需要更改设置。在您更改选项后，单击 Preview (预览)，以便更新嵌入代码。

12. 要在网页中嵌入 Flash 媒体播放，请复制 Preview Embed Code (预览嵌入代码) 框的值，然后将其粘贴到网站的 HTML 代码内。

    Note

    如果希望用户使用在您自己的域上托管的基于 Flash 的播放器来查看您的实时流，请参阅 如何为在我自己的域上托管的 Flash 流更新 crossdomain.xml？ (p. 178)。

通过 CloudFront 在 Apple 设备上播放实时 HLS 流

1. 通过以下网址打开 AWS CloudFormation 控制台：https://console.aws.amazon.com/cloudformation/。
2. 选择实时流堆栈。
3. 在底部窗格中，单击 Outputs (输出) 选项卡。
4. 复制 LiveHLSManifest 密钥的值，例如，`http://d123.cloudfront.net/hls-live/livepkgr/_definst_/liveevent/livestream.m3u8`。
5. 使用 iOS 设备导航至此 URL，以便验证 HLS 流是否正常工作。

有关在哪里使用 URL 以服务于各种 iOS 设备、QuickTime 和 Safari 的信息，请转至 iOS Developer Library 中的 HTTP 实时流概述。

有关发布和播放 HTTP 实时流的更多信息，请参阅 *Adobe Media Server 5.0.1 开发人员指南*中的用于发布和播放 HTTP 实时流的 URL。

下一步：删除 AWS CloudFormation 实时流堆栈 (p. 176)

删除 AWS CloudFormation 实时流堆栈

当您的实时事件结束时，请删除为实时流而创建的堆栈。这将删除为您的实时流事件而创建的 AWS 资源，并停止 AWS 资源计费。

删除 AWS CloudFormation 实时流堆栈

1. 登录 AWS 管理控制台，并通过以下网址打开 AWS CloudFormation 控制台：https://console.aws.amazon.com/cloudformation/。
2. 选择堆栈，然后单击 Delete Stack (删除堆栈)。
3. 单击 Yes, Delete (是，删除) 以确认。
4. 要跟踪堆栈的删除进度，请选择堆栈，然后单击底部框架中的 Events (事件) 选项卡。
5. 如果您近期不打算再次使用实时流，则可以取消对 Amazon EC2 上的 Adobe Media Server 的订阅。要取消订阅，请转至 Amazon Web Services 上的 Adobe Media Server 5，单击 Subscribe Now (立即订阅)，然后按照屏幕上的提示操作。

常见问题

我如何使用安全外壳 (SSH) 连接到运行 Adobe Media Server 5.0 的 Amazon EC2 实例？

Note

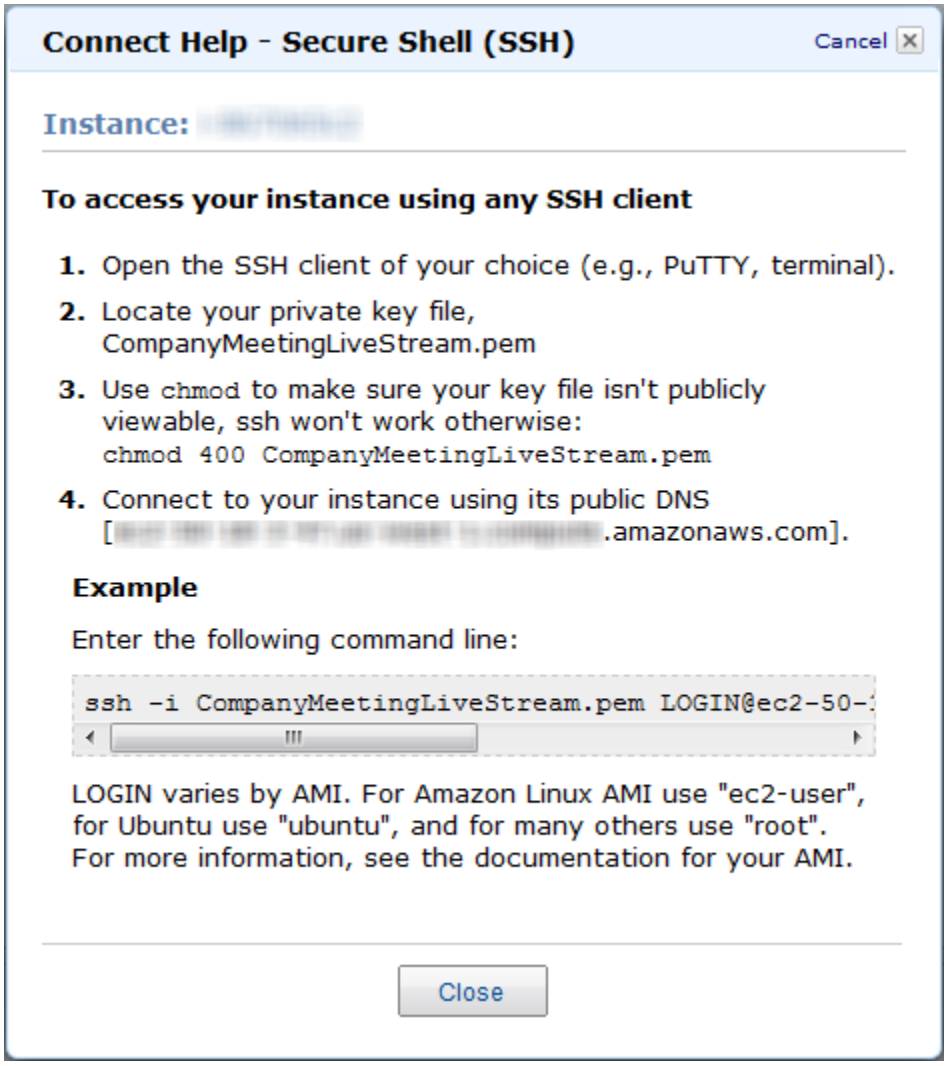
默认情况下，Amazon EC2 实例的 SSH 端口（端口 22）因安全原因而处于禁用状态。下面的步骤说明了如何启用 SSH 端口以及如何使用 SSH 连接到您的 Amazon EC2 实例。

在运行 Adobe Media Server 5.0 的 Amazon EC2 实例上启用对端口 22 的访问

1. 获得与您的 Amazon EC2 实例相关联的 Amazon EC2 安全组的名称：
   a. 登录 AWS 管理控制台，并通过以下网址打开 AWS CloudFormation 控制台：https://console.aws.amazon.com/cloudformation/。
   b. 在“Region (区域)”列表中，选择要在其中创建 Amazon EC2 实例的区域。
   c. 单击您的 AWS CloudFormation 堆栈所在的行。
   d. 在底部窗格中，单击 Resources (资源) 选项卡。
   e. 在 Stack Resources (堆栈资源) 表的左列，找到值为 `AMSOriginServerSecurityGroup` 的行。
   f. 在该行中，请记下 Physical ID (物理 ID) 列的值。
2. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
3. 在“Navigation (导航)”窗格中，单击 Security Groups (安全组)。
4. 在“Security Groups (安全组)”页面中，选择相应的行，其中，Name (名称) 列应该与您在步骤 1f 中获得的物理 ID 相匹配。
5. 在底部窗格中，单击 Inbound (入站) 选项卡。
6. 对于 Create a new rule (创建新规则)，请选择 SSH。
7. 单击 Add Rule (添加规则)。
8. 单击 Apply Rule Changes (应用规则更改)。

使用 SSH 连接到运行 Adobe Media Server 5.0 的 Amazon EC2 实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在“Navigation (导航)”窗格中，单击 Instances (实例)。
3. 右击正确的实例，然后单击 Connect (连接) 以查看关于如何使用 SSH 连接到 Amazon EC2 实例的说明。



如何为在我自己的域上托管的 Flash 流更新 crossdomain.xml？

您可以在创建 AWS CloudFormation 堆栈之前或之后更改 `crossdomain.xml` 中的权限：

- 如果您尚未创建 AWS CloudFormation 堆栈，请下载使用 Amazon CloudFront 和 Adobe Media Server 5.0 的实时流 AWS CloudFormation 模板（位于 https://s3.amazonaws.com/cloudfront-live/live-http-streaming-ams-5-0-1-using-cloudfront.txt）。在该模板中，编辑 `UserData` 部分（其中包含 `crossdomain.xml` 设置），并将更新后的模板保存在您的本地计算机上。然后使用更新后的模板创建您的 AWS CloudFormation 堆栈。
- 如果您已经创建 AWS CloudFormation 堆栈，请登录到运行 Amazon EC2 实例的 Adobe Media Server，然后更改跨域策略文件 `/mnt/webroot/crossdomain.xml` 中的权限。

有关编辑 `crossdomain.xml` 文件的更多信息，请参阅 Adobe 跨域策略文件规范。

使用 CloudFront 和 Adobe Media Server 5.0 的实时 HTTP 流的价格是多少？

除 Amazon EC2 上的 Adobe Media Server 每月产生 5.00 USD 的订阅费之外，您只需为所消耗的 AWS 资源付费：

- 有关 Amazon EC2 上运行的 Adobe Media Server 的定价信息，请参阅 Amazon Web Services 上的 Adobe Media Server 5/定价。
- 有关 CloudFront 的定价信息，请参阅 Amazon CloudFront 定价。

使用 AWS CloudFormation 不产生任何费用。

我如何为我的 Amazon EC2 实例或 CloudFront 分配创建别名记录 (CNAME) 别名？

您的运行 Adobe Media Server 5.0 的 Amazon EC2 实例同时附带一个内部和外部 DNS 名称。Amazon EC2 不提供修改这些 DNS 设置的访问权限。如果要将现有域名映射到您的运行 Adobe Media Server 的 Amazon EC2 实例，请使用 DNS 服务提供商，如 Amazon Route 53。当使用您自己的域名时，我们建议您使用别名记录 (CNAME) 映射到实例的外部 DNS 名称，而不是使用指向实例 IP 地址的 A 记录。

要将您自己的域名映射到 CloudFront 分配，请参阅 使用备用域名（别名记录）(p. 52)。

我如何连接到 Adobe Media Server 管理控制台？

连接到 Adobe Media Server 管理控制台

1. 登录 AWS 管理控制台，并通过以下网址打开 AWS CloudFormation 控制台：https://console.aws.amazon.com/cloudformation/。
2. 选择实时流堆栈。
3. 在底部窗格中，单击 Outputs (输出) 选项卡。
4. 复制 AMSAdminConsoleServerAddress 密钥的值。
5. 单击 AMSServerAdminConsole 密钥的值，例如，http://ec2-00-11-22-33.us-west-1.compute.amazonaws.com/ams_adminConsole.htm。
6. 在 Adobe Media Server 管理控制台的登录页面，在 Server Address (服务器地址) 中粘贴您在步骤 4 中复制的 AMSAdminConsoleServerAddress 密钥。
7. 在 Username (用户名) 和 Password (密码) 字段中，输入您在创建 AWS CloudFormation 实时流堆栈 (p. 168)中指定的值。
8. 单击 Login (登录)。

有关使用 Adobe Media Server 5.0 管理控制台的信息，请参阅 Adobe Media Server 文档。

Note

Adobe 建议您阻止所有对端口 1111 的外部访问，以便仅限您防火墙内的客户端访问管理控制台。作为替代方案，您可使用基于域的限制来限制访问服务器。有关更多信息，请参阅 Adobe 文档中的限制对 Adobe Media Administration Server 的访问。

禁用或限制对 Adobe Media Server 端口 1111 的访问

1. 获得与您的 Amazon EC2 实例相关联的 Amazon EC2 安全组的名称：

a. 登录 AWS 管理控制台，并通过以下网址打开 AWS CloudFormation 控制台：https://console.aws.amazon.com/cloudformation/。
b. 对于 Region (区域)，请单击您在其中创建 Amazon EC2 实例的区域名称。
c. 选择您的 AWS CloudFormation 堆栈所在的行。
d. 在底部窗格中，单击 Resources (资源) 选项卡。
e. 在 Stack Resources (堆栈资源) 表的 `AMSOriginServerSecurityGroup` 行中，请记下 Physical ID (物理 ID) 列的值。

2. 显示 Amazon EC2 控制台。
3. 在“Navigation (导航)”窗格中，单击 Security Groups (安全组)。
4. 在 Security Groups (安全组) 窗格中，选择 AWS CloudFormation 为您的 Amazon EC2 实例而创建的安全组。名称是您在步骤 1e 中记下的值。
5. 在底部窗格中，单击 Inbound (入站) 选项卡。
6. 要完全禁用对 Adobe Media Server 管理控制台的访问：

   a. 在 TCP Port (Service) (TCP 端口 (服务)) 列中，找到 1111。
   b. 在该行的 Action (操作) 列中，单击 Delete (删除)。
   c. 单击 Apply Rule Changes (应用规则更改)。

7. 要限制对选定 IP 地址的访问：

   a. 在 TCP Port (Service) (TCP 端口 (服务)) 列中，找到 1111，然后单击 Delete (删除)。
   b. 对于 Create a new rule (创建新规则)，请接受默认值 Custom TCP rule (自定义 TCP 规则)。
   c. 对于 Port range (端口范围)，请输入 1111。
   d. 对于 Source (源)，请输入一个 IP 地址或范围，或输入另一安全组的名称。有关更多信息，请单击 Help (帮助)。
   e. 单击 Add Rule (添加规则)。
   f. 要创建其他规则，请重复步骤 b 到步骤 e。
   g. 单击 Apply Rule Changes (应用规则更改)。

我是否可以将我的实时事件同时流式传输到 Apple 设备或兼容 Flash Player 的设备？

可以，Adobe Media Server 5.0 可以实现将实时流同时传送到基于 Flash 的设备和 iOS 设备。您可以流式传输到使用 HTML5 播放器或 Objective C（“原生”）应用程序的 Safari 浏览器。您还可使用适用于 iOS 的 Adobe AIR 在 iOS 设备上开发丰富的视频体验。

Adobe Media Server 5.0 是否支持 HTML5？

可以。Adobe Media Server 可使用 HLS 流格式将内容传送到 Apple iOS 设备上的 HTML5。对于支持 HTML5 的其他浏览器，您可以使用 Adobe Media Server 进行渐进式传送。

Adobe Media Server 中是否有日志记录？

可以。通过符合 W3C 标准的 ASCII 日志、实时使用情况监控器以及适用于服务器和流事件的完整 API，有助于确保您获得所有必要的工具，以便跟踪观众的内容使用情况并生成报告。有关监控和管理 Adobe Media Server 5.0 中日志文件的更多信息，请参阅 Adobe 文档中的监控和管理日志文件。

我如何在 Adobe Media Server 上启用身份验证？

在 AWS CloudFormation 为您的 Adobe Media Server Amazon EC2 实例而创建的安全组中，您可以限制对 RTMP 端口 1935（TCP 和 UDP）的访问。只需为端口 1935 创建新的 TCP 和 UDP 规则，然后删除端口 1935 的现有 TCP 和 UDP 规则，后者允许访问所有 IP 地址。

有关如何向安全组添加规则的简要概述，请参阅 我如何连接到 Adobe Media Server 管理控制台？ (p. 179)。有关 Amazon EC2 安全组的更多信息，请参阅 *Amazon Elastic Compute Cloud 用户指南* 中的 Amazon EC2 安全组。

HDS 和 HLS 相关文件上默认的缓存控制设置是什么？

HDS 和 HLS 相关文件上的默认缓存控制标头设置为以下值：

文件类型	**`Cache-Control`** 设置（秒）
.bootstrap	2
HDS 片段	60
.f4m	2
.m3u8	2
.ts	60

CloudFront 边缘缓存服务器遵守这些缓存控制标头。您可通过更改服务器上 `HttpStreamingF4MMaxAge`、`HttpStreamingBootstrapMaxAge` 和 `HttpStreamingFragMaxAge` 参数的值来更改默认设置。有关更多信息，请参阅 Adobe 文档中的 HTTP 流配置文件参考。

HLS 和 HDS 有什么区别？

HLS 是一种针对 Apple 设备而优化的文件容器格式。该容器支持 H.264/AAC 编码的视频和音频，并且基于 MPEG-2 传输流 (TS)。所有传送到 iOS（包括适用于 IOS 的 AIR）设备的视频都必须使用此格式。

HDS 是一种针对在 Flash Player 中运行的应用程序而优化的文件容器格式。该容器也支持 H.264/AAC 编码的视频和音频，并且基于 MPEG-4 TS。HDS 在适用于 iOS 的 AIR 上不受支持。

当流无法启动时，我如何排查 Amazon EC2 实例的问题？

如果您执行了步骤 验证 Adobe Media Server 是否正在运行 (p. 170)，但流仍未启动，请执行以下步骤，以确认 Amazon EC2 实例是否正常工作。

排查运行 Adobe Media Server 5.0 的 Amazon EC2 实例的问题

1. 在 AWS CloudFormation 控制台的顶部窗格里，选择堆栈。
2. 在底部窗格中，单击 Resources (资源) 选项卡。
3. 对于 AMSOriginServer 行，请记下 Physical ID (物理 ID) 列的值。
4. 转至 Amazon EC2 控制台。
5. 在 Region (区域) 列表中，选择在其中创建 AWS CloudFormation 堆栈的区域。
6. 在“Navigation (导航)”窗格中，单击 Instances (实例)。
7. 在 Instance (实例) 列中，找到您在步骤 c 中记下的值。
8. 选择对应的行。
9. 在底部窗格中，检查 Status Checks (状态检查) 选项卡上的信息，然后执行所建议的操作。

10. 返回步骤 验证 Adobe Media Server 是否正在运行 (p. 170)，然后重复步骤 2 到步骤 5。

在哪里可以找到使用 Adobe Flash Media Server 4.5 的实时流文档？

有关使用 Adobe Flash Media Server 4.5 的实时流文档，请参阅 CloudFront API 版本 2012-07-01 Amazon CloudFront 开发人员指南的“CloudFront 教程”一章中的“使用 CloudFront 和 Adobe Flash Media Server 4.5 的实时流”。

其他文档

Adobe 文档

- 使用 Amazon Web Services 上的 Adobe Media Server
- Adobe 跨域策略文件规范
- Flash Media Live Encoder
- Flash Media Live Encoder 常见问题
- Adobe Media Server 平台上的 HTTP 动态流视频编码和转码建议
- Adobe Media Server 5.0 技术概述

Amazon Web Services 文档

- Amazon Elastic Compute Cloud 文档
- AWS CloudFormation 文档

使用 Amazon CloudFront 和 IIS Media Services 4.1 的实时平滑流

Topics

使用 Amazon Web Services 的实时平滑流概述

平滑流是 Microsoft 的自适应流媒体技术实现，是一种基于 Web、采用标准 HTTP 的媒体内容传输形式。作为 IIS Media Services 的扩展，平滑流可以通过自适应流媒体技术将实时事件传输至 Microsoft Silverlight 等平滑流客户端。当您配置平滑流以使用 CloudFront 时，您将受益于 CloudFront 的全球 HTTP 网络规模以及根据延迟将观看者路由至网络上的边缘节点这一功能。要了解有关 CloudFront 的更多信息，请至 CloudFront 产品页面。

平滑流内容作为一系列 MPEG-4 (MP4) 片段交付至客户端，而这些片段可以在 CloudFront 边缘服务器上进行缓存。兼容平滑流的客户端使用特殊的探试程序，以动态监控当前网络和本地 PC 状况，并无缝切换客户端接收的平滑流演示的视频质量。当客户端播放片段时，网络状况可能发生变化（例如，带宽可能减少），或者，视频处理可能受在客户端上运行的其他应用程序的影响。客户端可立即从以不同比特率编码的流请求下一个片段，以便适应不断变化的状况。这使客户端能够流畅播放媒体内容，而不出现卡滞、缓冲或冻结。因此，用户可体验最高质量的无间断播放流。

要将现场直播编码成平滑流格式，您可使用 Microsoft Expression Encoder 4 Pro。要提供已编码的平滑流，您可以使用运行 Windows IIS Media Services 的 Amazon EC2 亚马逊系统映像 (AMI)。CloudFront 可以缓存实时视频和音频内容，而观看者可连接到 CloudFront 边缘服务器，以便使用诸如 Microsoft Silverlight 等兼容平滑流的客户端来播放流媒体。本教程将引导您完成整个设置过程。

Note

在运行 Windows IIS Media Services 的 Amazon EC2 亚马逊系统映像 (AMI) 中，不包含带 Service Pack 2 的 Microsoft Expression Encoder 4 Pro，并且不提供免费下载。有关功能和定价的信息，请转至 Microsoft 商店网站上的 Expression Encoder 4 Pro 页面。您还可使用第三方编码工具，对您的视频进行实时平滑流编码。有关提供编码软件的 Microsoft 合作伙伴列表，请参阅 Microsoft 网站 IIS Media Services 页面上的“Partners (合作伙伴)”选项卡。

Note

本教程概述了如何将 CloudFront 与在 Amazon EC2 实例上运行的 Microsoft 实时平滑流相集成。有关如何管理和保护 Amazon EC2 实例的更多信息，请参阅 Amazon EC2 文档。有关本教程中未涉及的 Microsoft 实时平滑流选项的更多信息，请参阅 Microsoft 文档 (p. 197)。

要设置使用 Amazon Web Services (AWS) 的实时平滑流，请查看*平滑流部署指南*中的 IIS 平滑流系统要求。然后执行以下几个部分中的步骤：

1. 创建 Amazon Web Services 账户 (p. 184)
2. 创建 Amazon EC2 密钥对 (p. 184)
3. 创建 AWS CloudFormation 实时平滑流堆栈 (p. 185)
4. 验证您的 Amazon EC2 Windows Server 实例是否正在运行 (p. 188)
5. 获得您的 Windows 密码 (p. 188)
6. 对实时流进行编码 (p. 190)
7. 查看实时平滑流 (p. 195)
8. 删除 AWS CloudFormation 实时平滑流堆栈 (p. 195)

有关常见问题，请参阅 常见问题 (p. 196)。

有关其他 Microsoft 和 AWS 文档的链接，请参阅 其他文档 (p. 197)。

创建 Amazon Web Services 账户

如果您已有 AWS 账户，请跳至创建 Amazon EC2 密钥对 (p. 184)。如果您没有 AWS 账户，请通过以下步骤创建一个。

Note

AWS 会在您创建账户时自动为该帐户注册所有服务。您只需为使用的服务付费。

如何创建 AWS 账户

1. 请转到 http://aws.amazon.com，然后单击 Create an AWS Account (创建 AWS 账户)。
2. 按照屏幕上的说明进行操作。

   作为注册流程的一部分，您会收到一个电话，需要您使用电话键盘输入一个 PIN 码。

下一步：创建 Amazon EC2 密钥对 (p. 184)

创建 Amazon EC2 密钥对

如果您在要配置实时平滑流的 Amazon EC2 区域中已有 Amazon EC2 密钥对，请跳至创建 AWS CloudFormation 实时平滑流堆栈 (p. 185)。如果您在该区域无密钥对，请执行以下步骤。

密钥对是一种类似于密码的安全证书。在本过程的后面部分，您将在创建 AWS CloudFormation 实时流堆栈时指定一个密钥对。在配置实时流后，您将使用此密钥对检索 Amazon EC2 Windows Server 实例的密码。

创建 Amazon EC2 密钥对

1. 登录 AWS 管理控制台，并通过以下网址打开 Amazon EC2 控制台：
   https://console.aws.amazon.com/ec2/。
2. 在“Region (区域)”列表中，单击要在其中创建密钥对的区域。

   您要创建密钥对的区域必须与要在本过程后面部分中创建 AWS CloudFormation 实时流堆栈的区域相同。我们建议您在距离实时事件最近的区域创建密钥对和实时流堆栈。
3. 在“Navigation (导航)”窗格中，单击 Key Pairs (密钥对)。
4. 在“Key Pairs (密钥对)”窗格中，单击 Create Key Pair (创建密钥对)。

5. 在“Create Key Pair (创建密钥对)”对话框中，输入密钥对的名称并记录下该名称。在实时流设置过程的后面部分中，您将在创建 AWS CloudFormation 实时流堆栈时输入此值。
6. 单击 Create (创建)，此时将显示“Opening <key_pair_name>.pem (正在打开 <key_pair_name>.pem)”对话框。
7. 将 .pem 文件保存到您计算机上的安全位置。
8. 单击 Close (关闭) 以关闭“Create Key Pair (创建密钥对)”对话框。

下一步：创建 AWS CloudFormation 实时平滑流堆栈 (p. 185)

创建 AWS CloudFormation 实时平滑流堆栈

下面的步骤将使用 AWS CloudFormation 模板来创建一个堆栈，用于启动实时平滑流所需的 AWS 资源，包括一个 Amazon EC2 实例。

Important

当您创建用于部署 Amazon EC2 实例的 AWS CloudFormation 堆栈时，将会开始产生按小时计算的 Amazon EC2 实例费用。无论您是否使用 Amazon EC2 实例来流式传输实时视频，费用都会一直累积，直到您删除 AWS CloudFormation 堆栈为止。有关更多信息，请参阅 Amazon Elastic Compute Cloud (Amazon EC2) 详情页面上的定价。当实时事件结束时，请删除为实时平滑流而创建的堆栈。这将删除为您的实时流事件而创建的 AWS 资源，并停止 AWS 资源计费。有关更多信息，请参阅 删除 AWS CloudFormation 实时平滑流堆栈 (p. 195)。

创建 AWS CloudFormation 实时流堆栈

1. 在下面的列表中，单击要在其中创建堆栈的 Amazon EC2 区域。此时将启动“Create Stack (创建堆栈)”向导，并会在 Provide a Template URL (提供模板 URL) 字段中自动输入一个区域特定的值。

   美国东部（弗吉尼亚）

   美国西部（俄勒冈）

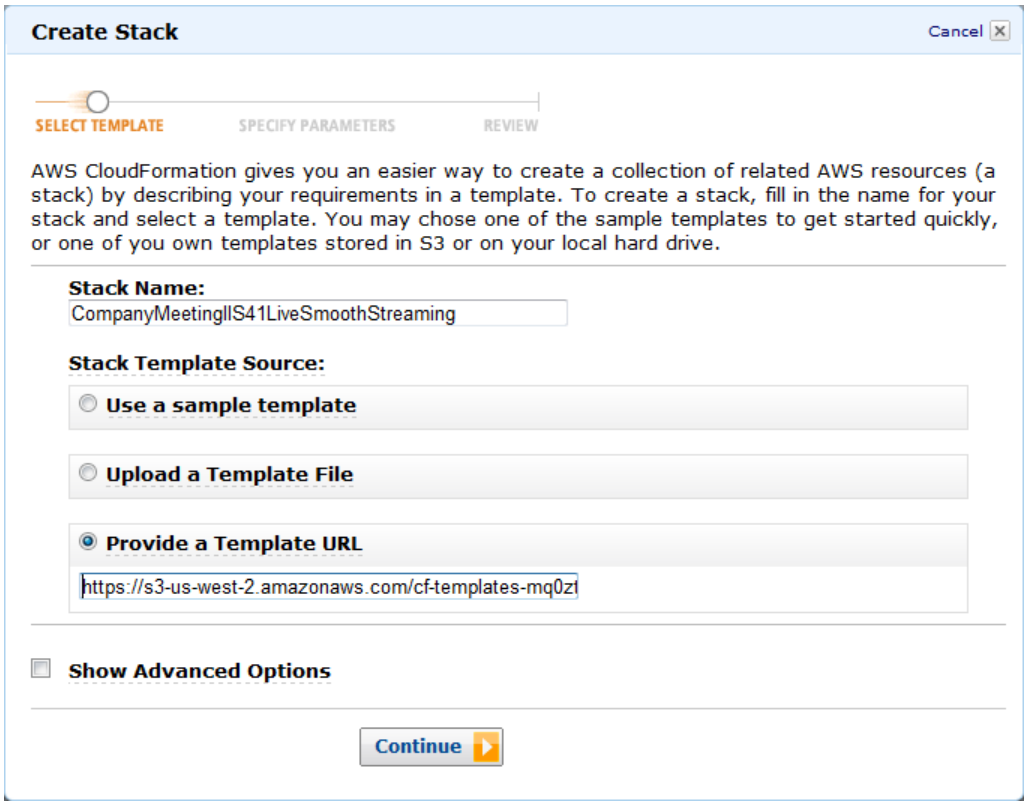
   美国西部（加利福尼亚北部）

   欧洲（爱尔兰）

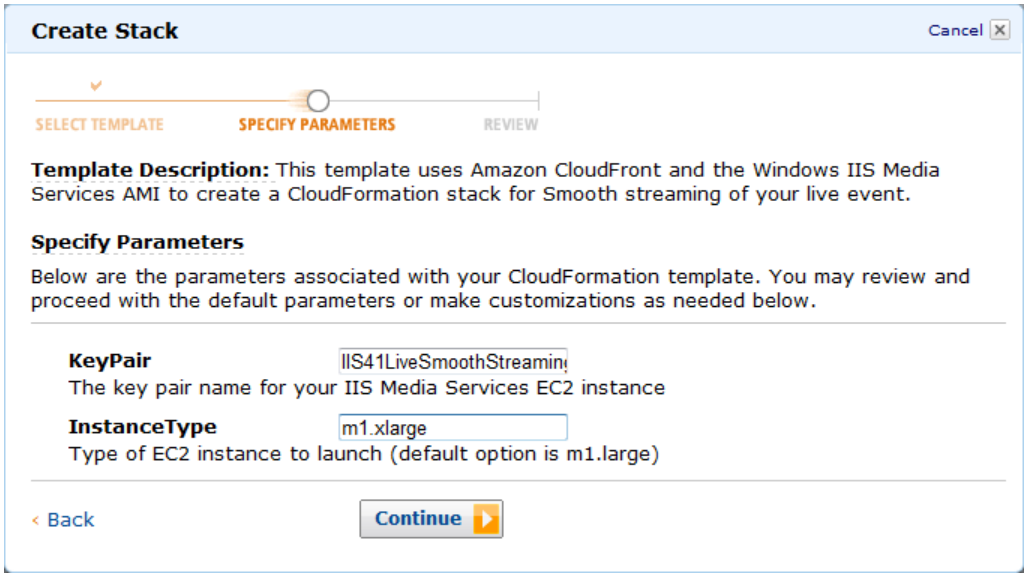
   亚太地区（新加坡）

   亚太地区（东京）

   南美洲（圣保罗）
2. 如果您尚未登录 AWS Management Console，请在出现提示时登录。
3. *可选：*在“Create Stack (创建堆栈)”向导中，更改 Stack Name (堆栈名称) 字段的值。堆栈名称不得包含空格，必须是 AWS 账户内的唯一名称。



4. 请勿更改 Stack Template Source (堆栈模板源) 选项或 Provide a Template URL (提供模板 URL) 的值。
5. *可选*：要配置 SNS 通知、要指定您愿意等待的堆栈创建时间以及要选择是否在堆栈创建失败后回滚更改，请选中 Show Advanced Options (显示高级选项) 复选框，然后指定适用的值。
6. 单击 Continue (继续)。

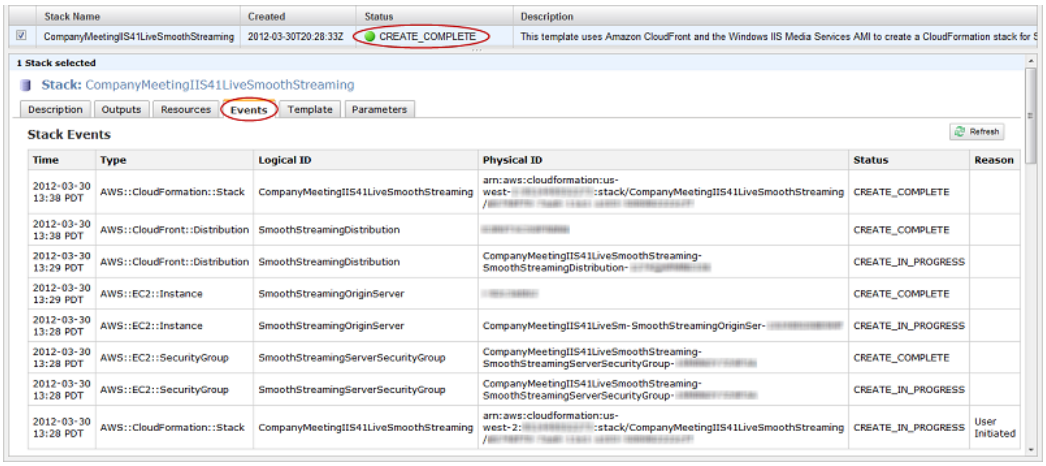


7. 在“Specify Parameters (指定参数)”页面的 KeyPair (密钥对) 字段中，在要创建实时流堆栈的区域输入 Amazon EC2 密钥对的名称。该密钥对必须与您当前登录的账户相关联。如果您在执行创建 Amazon EC2 密钥对 (p. 184)中的步骤时创建了密钥对，请输入该密钥对的名称。
8. 在 InstanceType 字段中，输入实例类型，然后单击 Continue (继续)。默认值为 *m1.xlarge*。

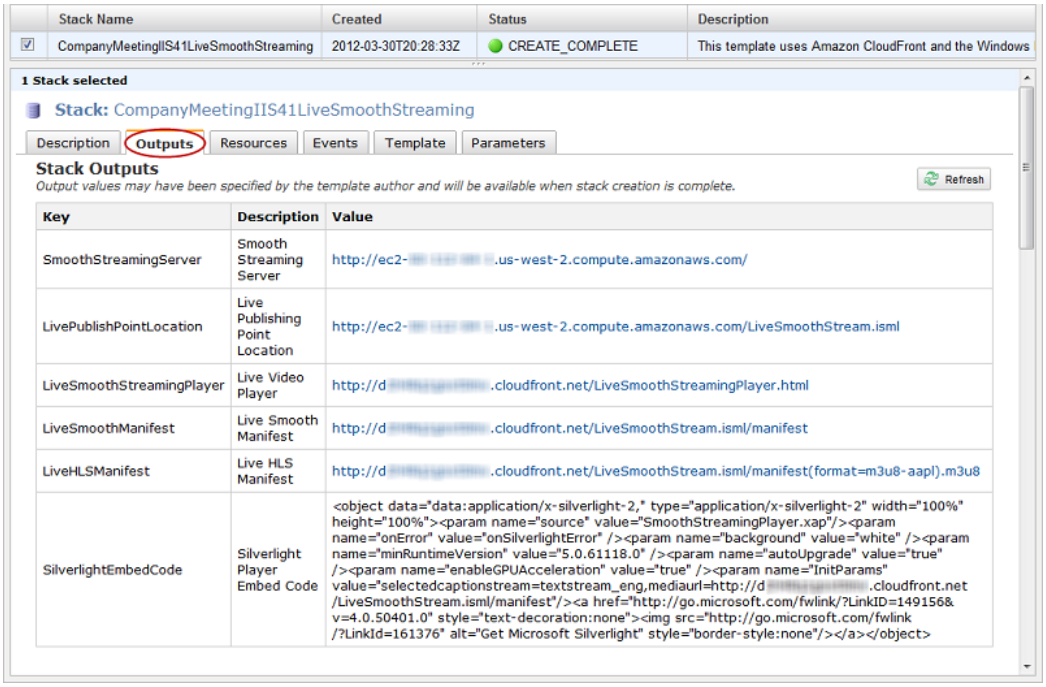
   实例类型决定了运行 Windows Server 的 Amazon EC2 实例的定价。有关适用于 Windows 的 Amazon EC2 实例类型的定价，请转至运行 Microsoft Windows Server 和 SQL Server 的 Amazon EC2。
9. 检查堆栈的设置。如果满意这些设置，请单击 Create Stack (创建堆栈)。

   堆栈创建可能需要花费几分钟。要跟踪堆栈的创建进度，请选择堆栈，然后单击底部框架中的 Events (事件) 选项卡。如果 AWS CloudFormation 无法创建堆栈，“Events (事件)”选项卡将列出错误消息。

   当您的堆栈准备就绪后，在顶部框架中，堆栈的状态将变为 CREATE_COMPLETE。



当您的堆栈完成创建后，请单击 Outputs (输出) 选项卡，然后将显示堆栈创建输出结果。当您在本过程后面部分中设置 Microsoft Expression Encoder 时，您将用到这些值。



下一步：验证您的 Amazon EC2 Windows Server 实例是否正在运行 (p. 188)

验证您的 Amazon EC2 Windows Server 实例是否正在运行

在 AWS CloudFormation 创建堆栈后，请执行以下步骤来验证 Windows IIS Media Services Web 服务器是否正在您通过 AWS CloudFormation 而配置的 Amazon EC2 实例上运行。

验证您的 Windows 服务器是否正在运行

1. 登录 AWS 管理控制台，并通过以下网址打开 AWS CloudFormation 控制台：https://console.aws.amazon.com/cloudformation/。
2. 在顶部窗格中，选择您在创建 AWS CloudFormation 实时平滑流堆栈 (p. 185)中创建的堆栈。
3. 在底部窗格中，单击 Outputs (输出) 选项卡。
4. 单击 SmoothStreamingServer 密钥的值，例如，http://ec2-00-11-22-33.us-west-1.compute.amazonaws.com。

   此时将显示 Windows IIS Server 横幅屏幕，说明您的 Windows Server 正在运行。

下一步：获得您的 Windows 密码 (p. 188)

获得您的 Windows 密码

要连接到运行 Windows Server 2008 R2 和 IIS Media Services 的 Amazon EC2 实例，请通过以下步骤检索 Windows Server 管理员账户的初始密码。您只需为您的 Amazon EC2 实例检索一次密码。完成此步骤后，您就可以使用您的 Amazon EC2 实例，就像任何 Windows Server 计算机一样。

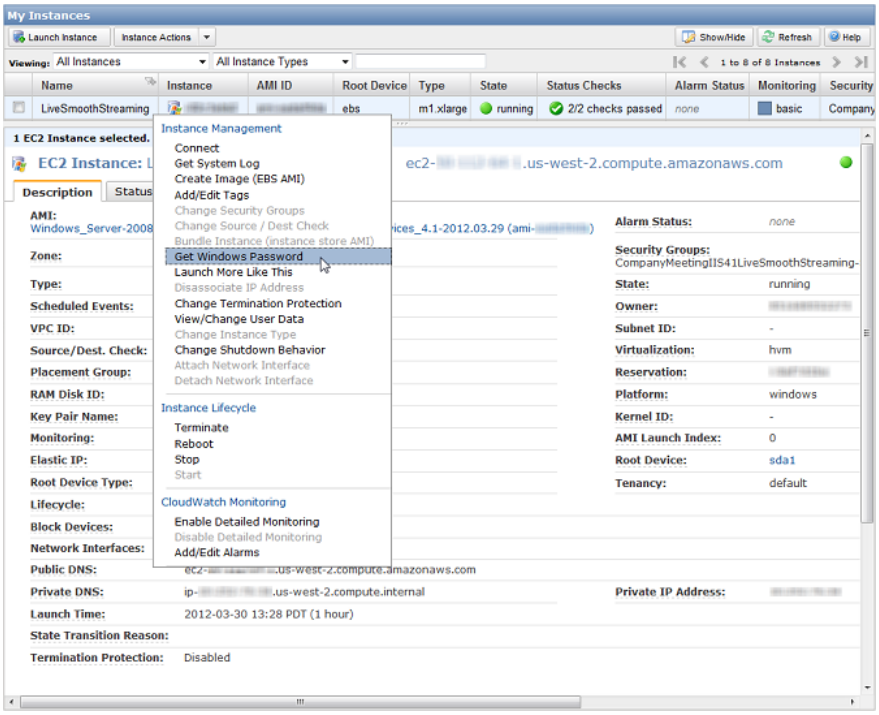
有关如何连接到运行 Windows 的 Amazon EC2 实例的更多信息，请转至入门指南：适用于 Windows 的 AWS 计算基础知识。

Important

Amazon EC2 最长需要 30 分钟来从 Windows Server 检索您的密码。

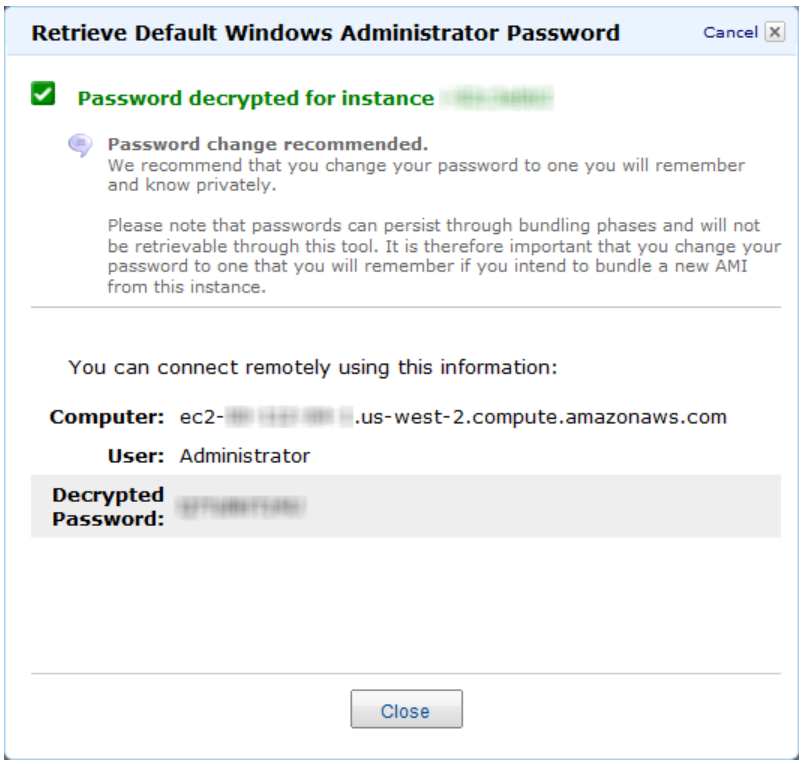
检索 Amazon EC2 实例的 Windows 密码

1. 确认您可以访问自己在创建 Amazon EC2 密钥对 (p. 184)中创建的 Amazon EC2 私有密钥文件（.pem 文件）。
2. 登录 AWS 管理控制台，并通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
3. 在 Region (区域) 列表中，单击要在其中创建实时平滑流 Amazon EC2 实例的区域。
4. 在 Navigation (导航) 窗格中，单击 Instances (实例)。
5. 在“My Instances (我的实例)”窗格中，右键单击 Name (名称) 列的值为 LiveSmoothStreaming 的实例，然后单击 Get Windows Password (获取 Windows 密码)。



6. 在“Retrieve Default Windows Administrator Password (检索默认 Windows 管理员密码)”页面上，单击 Browse (浏览)，然后浏览到您的计算机上保存 .pem 文件的位置。
7. 选择 .pem 文件，文件内容将会显示在窗口中。

8. 单击 Decrypt Password (解密密码)。
9. 记下密码。您需要使用此密码连接到 Amazon EC2 实例。



10. *可选但推荐：*登录到您刚才启动的 Windows Server 实例，并更改默认 Windows Server 账户的密码。用户名是 Administrator。

    您可能还需要创建另一个用户账户并将其添加到管理员组。另一个管理员账户是一种保护措施，以防止您万一忘记您的管理员密码或管理员账户出现问题。

Note

有关如何针对您的 Windows 服务器更新 Amazon EC2 安全组设置，以便您使用端口 3389 来访问服务器的信息，请参阅我如何启用对 Windows 服务器的访问？ (p. 196)。有关如何使用管理员账户登录实例的信息，请参阅我如何安全地连接到运行 Windows IIS Media Services 的 Amazon EC2 实例？ (p. 197)。

下一步：对实时流进行编码 (p. 190)

对实时流进行编码

通过本部分中的步骤创建一个使用 Microsoft Expression Encoder 4 Pro SP2 的现场直播项目，并将您的实时流发布到运行 Windows Server 和 Windows IIS Media Services 的 Amazon EC2 实例上的实时平滑流发布点。

要了解有关使用 Microsoft Expression Encoder 的现场直播的更多信息，请转至 Microsoft Expression 网站上的 Creating a Live Broadcasting Project (创建现场直播项目)。

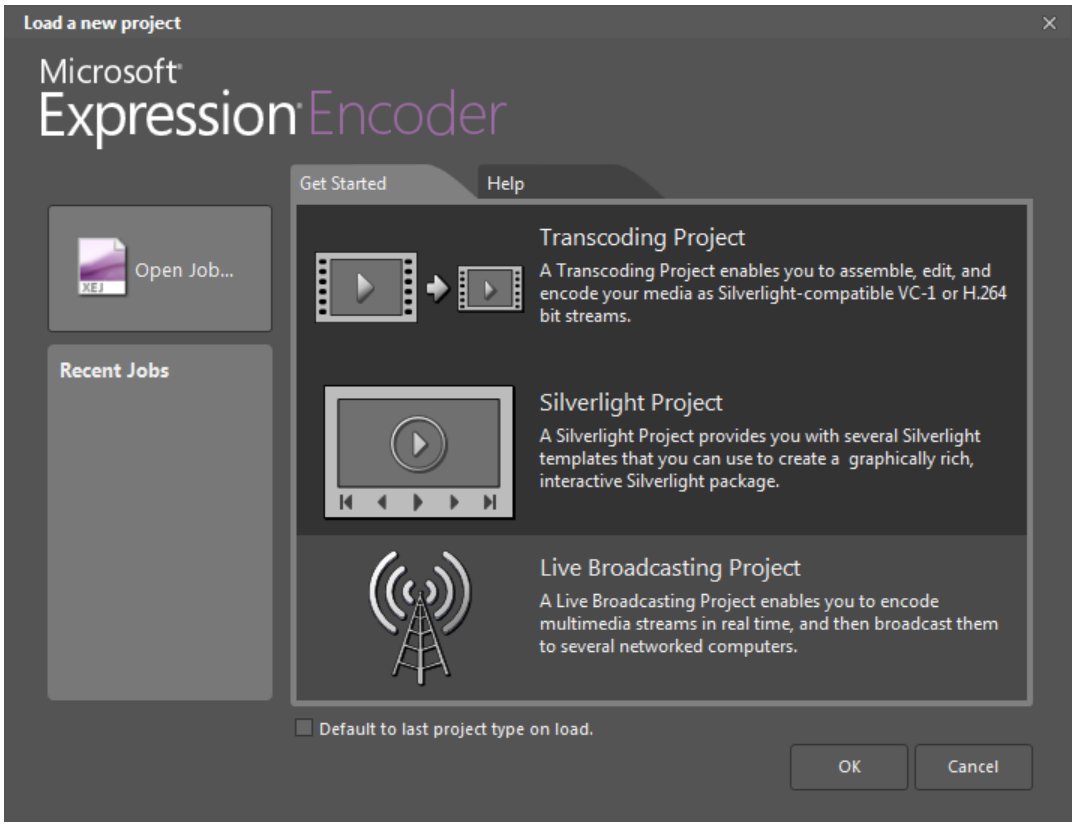
Note

带 Service Pack 2 的 Microsoft Expression Encoder 4 Pro 不提供免费下载。有关功能和定价的更多信息，请转至 Microsoft 商店网站上的 Expression Encoder 4 Pro 页面。

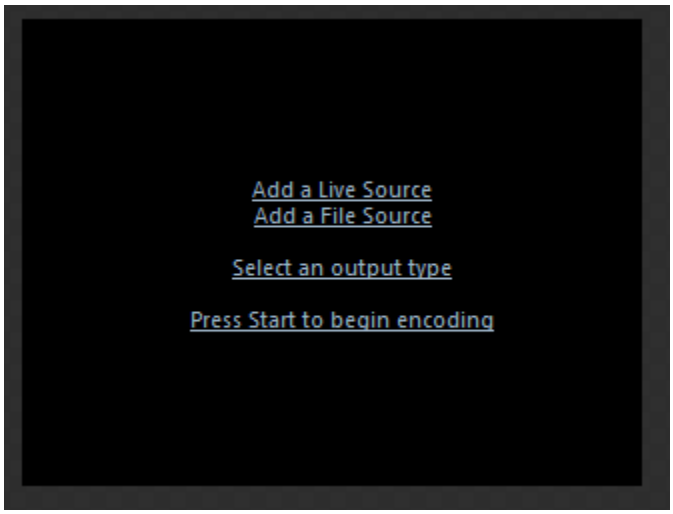
您还可使用第三方编码工具，对您的视频进行实时平滑流编码。有关提供编码软件的 Microsoft 合作伙伴列表，请参阅 Microsoft 网站 IIS Media Services 页面上的“Partners (合作伙伴)”选项卡。

对现场直播进行编码

1. 登录到您将用来广播实时流的计算机。
2. 在 Windows 开始菜单上，单击 All Programs (所有程序) > Microsoft Expression > Microsoft Expression Encoder 4。
3. 在 Load a new project (加载新项目) 对话框中，单击 Live Broadcasting Project (现场直播项目)。



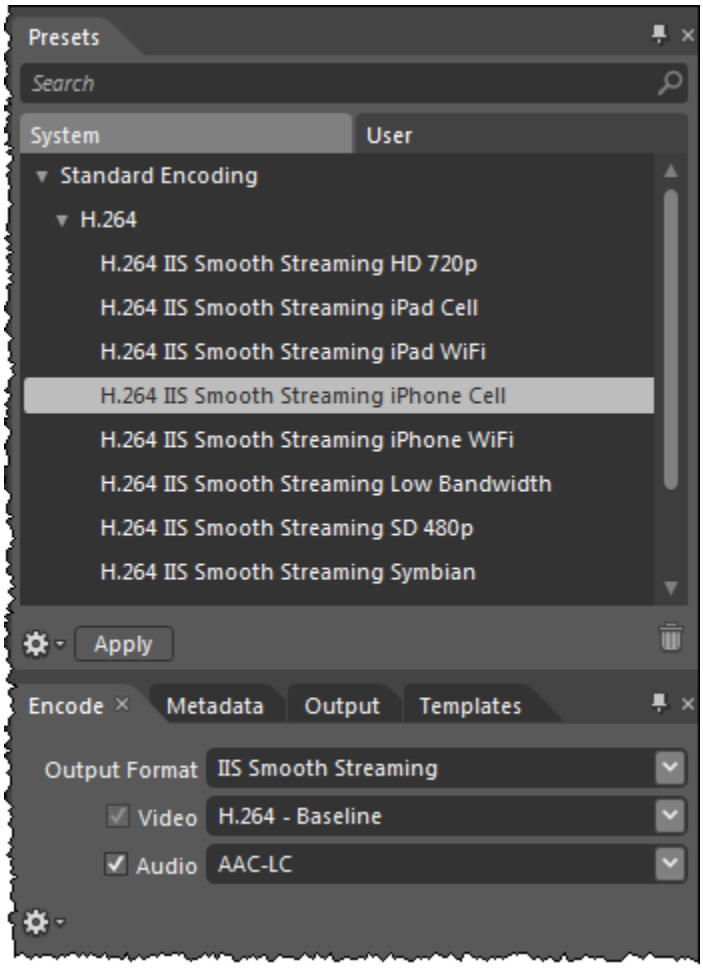
4. 单击 Add a Live Source (添加实时源) 以用于您的现场直播。



Note

您可以连接多个摄像设备，如 USB 摄像头或 FireWire (IEEE 1394) 数字摄像机。虽然您可连接多个实时源，但一次只能流式传输一个。有关设置现场直播源的更多信息，请转至 Microsoft Expression 网站上的 Set Live Sources (设置实时源)。

5. 在 Presets (预设) 选项卡上，选择支持您的实时平滑流场景比特率和编码要求的编码预设。选择名称里有 IIS Smooth Streaming (IIS 平滑流) 的选项。



当您单击 Apply (应用) 时，Encode (编码) 选项卡上的 Output Format (输出格式)、Video (视频) 和 Audio (音频) 设置将自动更新为您所选择的编码预设中的值。

有关预设的更多信息（例如，输出的流数量和使用的编解码器），请将鼠标指针悬停于预设名称上。

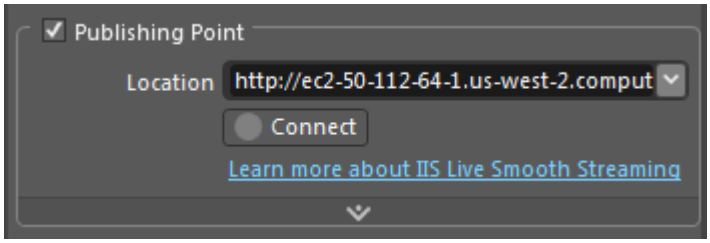
Note

或者，您可以在 Encode (编码) 选项卡上指定自定义设置。有关更多信息，请转至 Microsoft Expression 网站上的以下主题：

- 设置输出格式
- 视频设置
- 音频设置

6. 在 Microsoft Expression Encoder 中，单击 Output (输出) 选项卡。
7. 在 Output (输出) 选项卡上，选中 Publishing Point (发布点) 复选框。
8. 登录 AWS 管理控制台，并通过以下网址打开 AWS CloudFormation 控制台：https://console.aws.amazon.com/cloudformation/。
9. 在 AWS CloudFormation 控制台的底部窗格中，单击 Outputs (输出) 选项卡。
10. 复制 LivePublishPointLocation 密钥的值，例如，http://ec2-00-11-22-33.us-west-1.compute.amazonaws.com/LiveSmoothStream.isml。

11. 在 Microsoft Expression Encoder 中，将上一步中复制的 URL 粘贴到 Location (位置) 字段。



12. 单击 Connect (连接)，以启动一个到您 Windows 服务器上发布点的连接。
13. 当提示您输入发布点管理员密码时，请输入以下值：

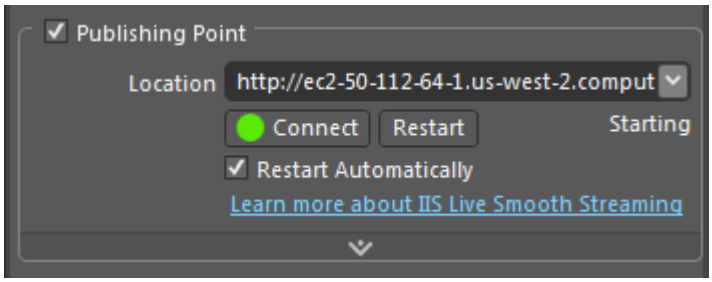
    - User Name: (用户名:)Administrator
    - Password: (密码:)：您在获得您的 Windows 密码 (p. 188)中检索到的 Windows Server 密码。

    然后单击 OK (确定)。

    Note

    系统已为您 Windows 服务器上的默认网站配置 Windows 身份验证，以便您可从 Microsoft Expression Encoder 4 Pro SP2 连接到服务器上的实时发布点。要了解有关 Windows 身份验证的更多信息，请转至 IIS 网站。要了解 IIS 7 中的可用身份验证机制，请转至 Microsoft 网站。

14. 当成功建立连接后，发布点状态将变为 Starting (启动)。此外，Connect (连接) 按钮旁将显示一个 Restart (重启) 按钮，就位于“Publishing Point (发布点)”部分中 Location (位置) 字段的下方。



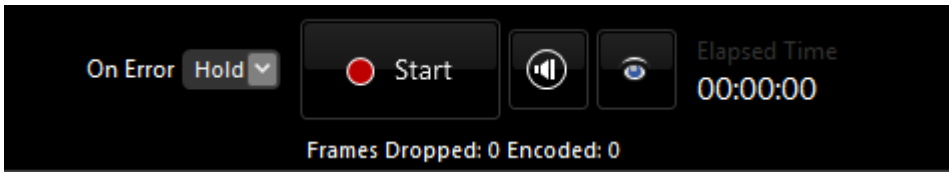
    Note

    “Starting (开始)”状态意味着发布点已准备好接收实时流。当实时源连接到发布点并开始推送内容时，“Starting (开始)”状态将变为“Started (已开始)”，意味着发布点正在接收实时流。

    Note

    带 SP2 的 Microsoft Expression Encoder 4 Pro 利用 Windows IIS Media Services 4.1 所包含的 REST API，以便帮助您管理 Windows 服务器上的实时发布点。有关更多信息，请转至 IIS 博客。

15. 单击 Start (开始)，以便开始对您的现场直播进行编码，并将其发布到运行 Windows Server 和 IIS Media Services 的 Amazon EC2 实例上的发布点。



当广播运行时，您可以在相应的面板中监控统计数据和连接数据。有关如何监控这些数据的更多信息，请参阅 Microsoft Expression 网站上的以下主题：

- 使用“Statistics (统计)”面板
- 使用“Connections (连接)”面板

下一步：查看实时平滑流 (p. 195)

查看实时平滑流

请执行以下步骤，以便查看使用 CloudFront 的实时平滑流。您还可以在自己的网页中嵌入 Microsoft Silverlight 播放器代码。

1. 登录 AWS 管理控制台，并通过以下网址打开 AWS CloudFormation 控制台：https://console.aws.amazon.com/cloudformation/。
2. 选择实时流堆栈。
3. 在 AWS CloudFormation 控制台的底部窗格中，单击 Outputs (输出) 选项卡。
4. 单击 LiveSmoothStreamingPlayer 密钥的值，例如，http://d123.cloudfront.net/LiveSmoothStreamingPlayer.html。
5. 要将 Silverlight 播放器代码嵌入到您的网页中，请在 Outputs (输出) 选项卡上，复制 SilverlightEmbedCode 密钥的值。

   Note

   Microsoft 建议观看者安装 Microsoft Silverlight 的最新版本，以获得最佳播放体验。

6. 要在 iPad 或 iPhone 等 Apple 设备上观看您的实时流，请从兼容的 Apple 设备中显示 AWS CloudFormation 控制台，然后单击 LiveHLSManifest 密钥的值。清单 URL 类似于 http://d123.cloudfront.net/LiveSmoothStream.isml/manifest(format=m3u8-aapl).m3u8。

   有关在哪里使用 URL 以服务于各种 iOS 设备、QuickTime 和 Safari 的信息，请转至 iOS Developer Library 中的 HTTP Live Streaming Overview。

下一步：删除 AWS CloudFormation 实时平滑流堆栈 (p. 195)

删除 AWS CloudFormation 实时平滑流堆栈

当实时事件结束时，请删除为实时平滑流而创建的堆栈。这将删除为您的实时事件而创建的 AWS 资源，并停止 AWS 资源计费。

删除 AWS CloudFormation 实时流堆栈

1. 登录 AWS 管理控制台，并通过以下网址打开 AWS CloudFormation 控制台：https://console.aws.amazon.com/cloudformation/。
2. 选中该堆栈的复选框，然后单击 Delete Stack (删除堆栈)。

3. 单击 Yes, Delete (是，删除) 以确认。
4. 要跟踪堆栈的删除进度，请选中该堆栈的复选框，然后单击底部框架中的 Events (事件) 选项卡。

常见问题

使用 CloudFront 的实时平滑流的价格是多少？

要平滑流式传输您的实时事件，您仅需为所消耗的 AWS 资源付费：

- 有关运行 Windows Server 的 Amazon EC2 实例的定价信息，请参阅运行 Microsoft Windows Server 和 SQL Server 的 Amazon EC2 页面上的 Pricing (定价)。
- 有关 CloudFront 的定价信息，请参阅 Amazon CloudFront 定价。

使用 AWS CloudFormation 不产生任何费用。

我是否可以将我的实时流视频同时传送到平滑流客户端和 Apple 设备？

可以。您可以使用 Microsoft Expression Encoder 4 Pro，以便同时针对平滑流客户端（例如，Microsoft Silverlight）和 Apple 设备（例如，iPad 和 iPhone）对您的实时视频进行编码。当您的 AWS CloudFormation 堆栈启动后，您将在 AWS CloudFormation 模板的 Outputs (输出) 选项卡上找到实时平滑流 (.isml) 和 Apple HLS (.m3u8) 清单文件 URL。

我可以如何为我的 Amazon EC2 实例或我的 CloudFront 分配设置别名记录 (CNAME) 别名？

您的 Amazon EC2 Windows Server 实例同时附带一个内部和外部 DNS 名称。Amazon EC2 不提供修改这些 DNS 设置的访问权限。如果要将现有域名映射到您的运行 Windows Server 的 Amazon EC2 实例，请使用 DNS 服务提供商，如 Amazon Route 53。当使用您自己的域名时，我们建议您使用别名记录 (CNAME) 映射到实例的外部 DNS 名称，而不是使用指向实例 IP 地址的 A 记录。

要将您自己的域名映射到 CloudFront 分配，请参阅 使用备用域名（别名记录） (p. 52)。

我如何启用对 Windows 服务器的访问？

在 Windows 服务器上通过选定的 IP 地址启用对端口 3389 的访问

默认情况下，您的 Windows 服务器实例的 Amazon EC2 安全组未启用端口 3389；这是您用来管理 Windows 服务器的端口。如果您要登录自己的 Windows 服务器实例，请执行以下步骤以启用通过端口 3389 的访问。

1. 登录 AWS 管理控制台，并通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。

2. 在 Region (区域) 列表中，单击要在其中使用 AWS CloudFormation 创建 Amazon EC2 实例的 Amazon EC2 的区域。
3. 在 Navigation (导航) 窗格中，单击 Security Groups (安全组)。
4. 在 Security Groups (安全组) 窗格中，单击 Name (名称) 列的值以您在创建 AWS CloudFormation 实时平滑流堆栈 (p. 185)中创建的 AWS CloudFormation 堆栈名称开头的行。
5. 在底部窗格中，单击 Inbound (入站) 选项卡。
6. 启用对 Windows 服务器的访问并指定能够访问该服务器的客户端 IP 地址：

   a. 在 Create a new rule (创建新规则) 列表中，请勿更改默认值 Custom TCP rule (自定义 TCP 规则)。
   b. 在 Port range (端口范围) 字段中，输入 3389。
   c. 在 Source (源) 字段中，输入一个 IP 地址或范围，或输入另一安全组的名称。有关更多信息，请单击 Help (帮助)。
   d. 单击 Add Rule (添加规则)。
   e. 要创建其他规则，请重复步骤 a 到步骤 d。
   f. 单击 Apply Rule Changes (应用规则更改)。

我如何安全地连接到运行 Windows IIS Media Services 的 Amazon EC2 实例？

要连接到 Windows 服务器实例，您必须检索管理员账户的初始密码，然后与 Windows 远程桌面一起使用。您还需要您创建的私有密钥文件的内容，例如，*<keypairname.pem>*.pem。有关更多信息，请转至入门指南：适用于 Windows 的 AWS 计算基础知识。

我如何限制从另一个域对实时平滑流内容的访问？

Microsoft Silverlight 包括跨域连接支持，这使 Silverlight 播放器可以从平滑流内容起始位置之外的位置访问内容。Silverlight 中的安全策略系统要求必须从目标域下载一个名为 `ClientAccessPolicy.xml` 的 Silverlight 策略文件，然后才允许网络连接访问该目标域下的网络资源。在 Amazon EC2 上运行的 Windows 服务器中，默认网站的根目录下已包含一个默认策略文件。要限制跨域访问，请登录您的 Windows 服务器并更新 `ClientAccessPolicy.xml` 文件。

其他文档

Microsoft 文档

- IIS 平滑流部署指南
- IIS Media Services 4.1 自述文件
- IIS 平滑流管理 REST 服务
- 在 IIS 7 中配置身份验证
- Microsoft Expression Encoder 博客
- 管理 Microsoft Expression Encoder 4 Pro SP2 的实时发布点
- Expression Encoder 4 Pro 中的实时 IIS 平滑流
- 采用 IIS Media Services 的 Apple HTTP 实时流

Amazon Web Services 文档

- 运行 Microsoft Windows Server 和 SQL Server 的 Amazon EC2

- Amazon Elastic Compute Cloud Microsoft Windows 指南
- Amazon CloudFront
- AWS CloudFormation

使用 Wowza Media Server 3.6 的实时 HTTP 流

您可以使用 Wowza Media Server 3.6 来创建实时流会话，以便使用 CloudFront 进行全球交付。Wowza Media Server 3.6 支持以下基于 HTTP 的流协议：

- HLS（HTTP 实时流）
- HDS（HTTP 动态流）
- 平滑流

当用户使用上述协议之一流式传输视频时，该视频将被分成一些较小的文件块，这些文件块会在 CloudFront 网络中进行缓存以提高性能和可扩展性。

本教程介绍了如何将 CloudFront 与 Amazon EC2 实例上运行的 Wowza Media Server 3.6 相集成。有关如何管理和保护 Amazon EC2 实例的更多信息，请参阅 Amazon EC2 文档。有关本教程中未包含的 Wowza Media Server 选项的更多信息，请参阅 Wowza 文档。

Topics

创建 Amazon Web Services 账户

如果您已有 AWS 账户，请跳至创建 Amazon EC2 密钥对 (p. 199)。如果您没有 AWS 账户，请通过以下步骤创建一个。

如何创建 AWS 账户

1. 转到 http://aws.amazon.com，然后单击 Sign Up (注册)。
2. 按照屏幕上的说明进行操作。

   作为注册流程的一部分，您会收到一个电话，需要您使用电话键盘输入一个 PIN 码。

下一步：创建 Amazon EC2 密钥对 (p. 199)

创建 Amazon EC2 密钥对

如果您在要配置实时流的 Amazon EC2 区域中已有 Amazon EC2 密钥对，请跳至获取 Wowza Media Server 3.6 许可证 (p. 200)。如果您在该区域无密钥对，请执行以下步骤。

密钥对是一种类似于密码的安全证书，并且它特定于具体的 AWS 区域。当您在本过程后面部分中为实时流创建 AWS CloudFormation 堆栈时，您需要指定密钥对。在配置实时流后，您可使用密钥对安全地连接到 Amazon EC2 实例。

创建 Amazon EC2 密钥对

1. 登录 AWS 管理控制台，并通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在右上角，单击要在其中创建密钥对的区域。

   您要创建密钥对的区域必须与要在本过程后面部分中创建 AWS CloudFormation 实时流堆栈的区域相同。我们建议，在创建密钥对和实时流堆栈时，应选择距离要发布实时流的位置最近的区域。
3. 在左侧导航窗格中，单击 Key Pairs (密钥对)。
4. 在“Key Pairs (密钥对)”窗格中，单击 Create Key Pair (创建密钥对)。
5. 在 Create Key Pair (创建密钥对) 对话框中，输入密钥对的名称并记录下该名称。您稍后在创建 AWS CloudFormation 实时流堆栈时将需要该名称。
6. 单击 Create (创建)，当系统提示时，请将 .pem 文件保存到您计算机上的安全位置。请注意，您将无法重新下载该文件。
7. 单击 Close (关闭) 以关闭 Create Key Pair (创建密钥对) 对话框。

下一步：获取 Wowza Media Server 3.6 许可证 (p. 200)

获取 Wowza Media Server 3.6 许可证

您需要 Wowza Media Server 3.6 许可证才能配置实时流。在 Wowza 网站上，您可以获得 30 天评估许可证或直接购买许可证：

- 获得 30 天评估许可证：转至 Get Your Free 30 Day Trial（免费试用 30 天）页面，然后按照说明获得您的许可证密钥。
- 购买许可证：转至 Licenses Built for You (为您构建的许可证) 页面，选择最适合您的许可选项，然后按照屏幕上的说明操作。

下一步：通过 AWS Marketplace 订阅 Wowza Media Server 3.6 (p. 200)

通过 AWS Marketplace 订阅 Wowza Media Server 3.6

下一步是在 AWS Marketplace 中订阅 Wowza Media Server 3.6。

为 Amazon Web Services 订购 Wowza Media Server

1. 转至 Amazon Web Services 页面，然后使用您的 Amazon.com 账户登录或创建一个新账户。
2. 转至 https://aws.amazon.com/marketplace/pp/B00ETDMYB2。检查详细信息，然后单击 Continue (继续)。
3. 单击 Launch with EC2 Console (使用 EC2 控制台启动) 选项卡。
4. 检查定价信息，然后单击 Accept Terms (接受条款)。

Important

请勿使用此页面上的按钮启动 Wowza。

下一步：创建 AWS CloudFormation 实时流堆栈 (p. 201)

创建 AWS CloudFormation 实时流堆栈

下面的步骤将使用 AWS CloudFormation 模板来创建一个堆栈，用于启动实时流所需的 AWS 资源，包括一个 Amazon EC2 实例。

Important

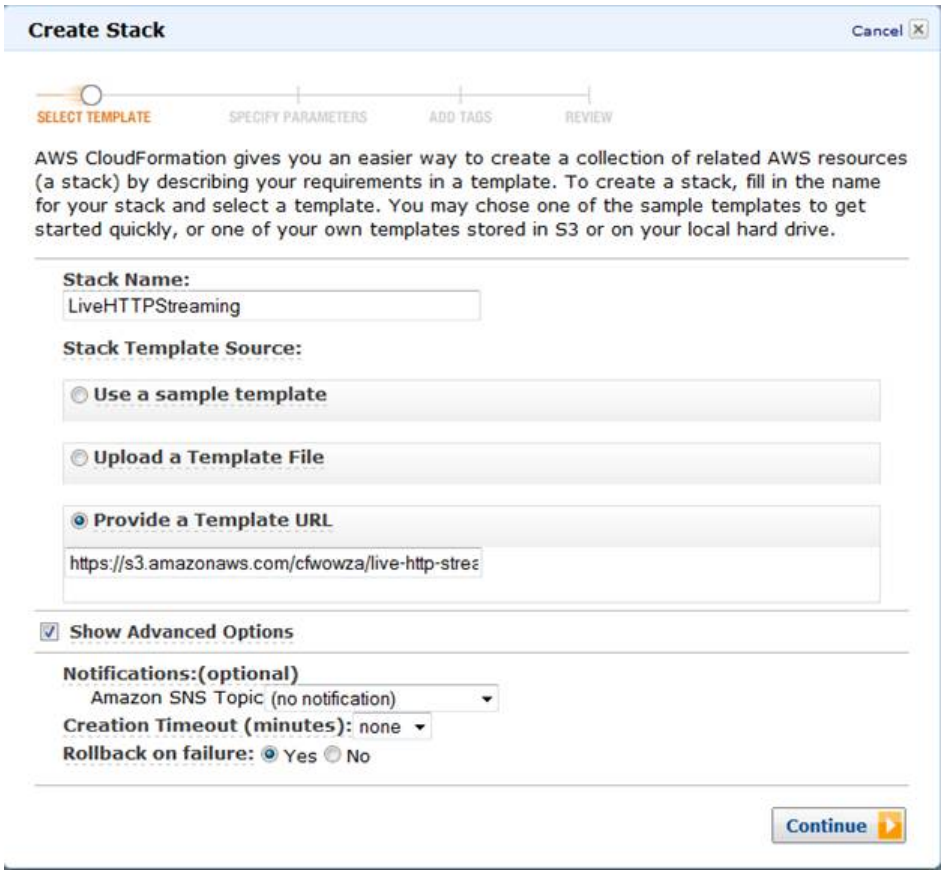
当您创建用于部署 Amazon EC2 实例的 AWS CloudFormation 堆栈时，将会开始产生按小时计算的实例费用。无论您是否使用 Amazon EC2 实例来流式传输实时视频，费用都会一直累积，直到您删除 AWS CloudFormation 堆栈为止。当您的实时事件结束时，请删除为实时流而创建的堆栈。这将删除为您的实时流事件而创建的 AWS 资源，并停止 AWS 资源计费。有关更多信息，请参阅 删除 AWS CloudFormation 实时流堆栈 (p. 207)。

创建 AWS CloudFormation 实时流堆栈

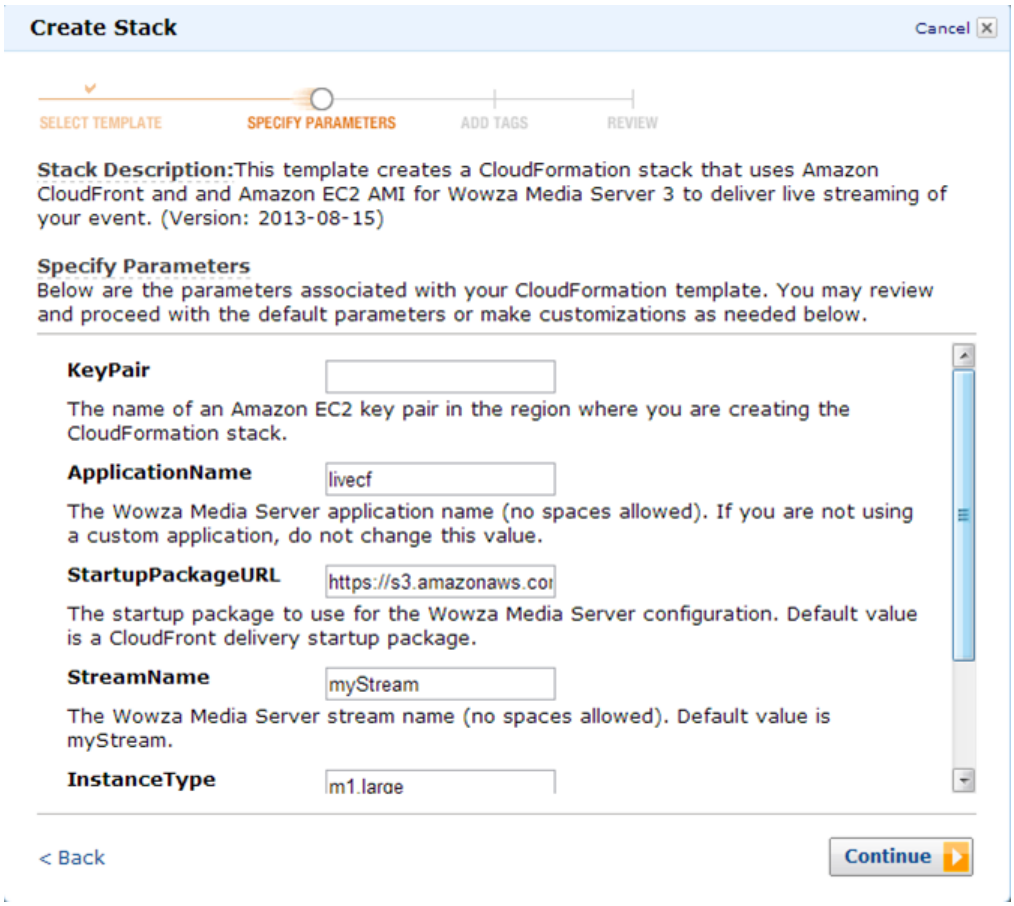
1. 要启动向导，请单击以下 Amazon EC2 区域之一：

   - 在美国东部（弗吉尼亚北部）创建堆栈
   - 在美国西部（俄勒冈）创建堆栈
   - 在美国西部（加利福尼亚北部）创建堆栈
   - 在欧洲（爱尔兰）创建堆栈
   - 在亚太地区（新加坡）创建堆栈
   - 在亚太地区（东京）创建堆栈
   - 在南美洲（圣保罗）创建堆栈

2. 如果您尚未登录 AWS Management Console，请在出现提示时登录。此向导将会启动，并且 Provide a Template URL (提供模板 URL) 下方将自动显示选定的 URL。
3. （可选）在 Create Stack (创建堆栈) 向导中，您可将 Stack Name (堆栈名称) 更改为适合您的实时流事件的值。堆栈名称不得包含空格，必须是 AWS 账户内的唯一名称。



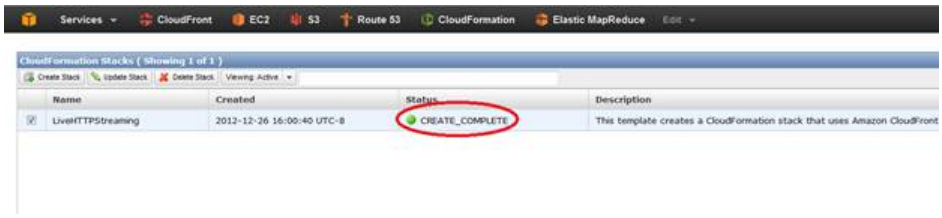
4. 将 Template (模板) 和 Provide a Template URL (提供模板 URL) 保持原样。
5. （可选）要配置 SNS 通知、要指定您愿意等待的堆栈创建时间以及要选择是否在堆栈创建失败后回滚更改，请选中 Show Advanced Options (显示高级选项)，然后指定所需的设置。
6. 单击 Continue (继续)。
7. 对于 KeyPair (密钥对)，请输入您要在其中创建实时流堆栈的区域的 Amazon EC2 密钥对名称。该密钥对必须与您当前登录的账户相关联。如果您在执行创建 Amazon EC2 密钥对 (p. 199)中的步骤时创建了密钥对，请在此处输入该名称。



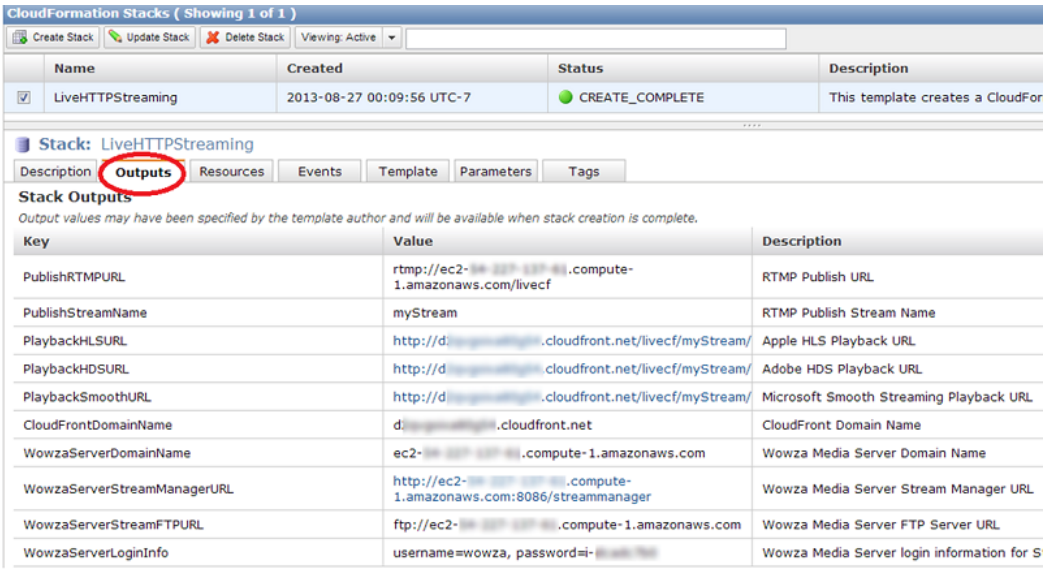
8. 对于 ApplicationName，请为您的 Wowza 应用程序输入一个短名（无空格）或保留默认值。
9. 对于 StartupPackageURL，请输入一个指向启动程序包、用于根据需要配置 Wowza Media Server 的 URL，或保留默认值。
10. 对于 StreamName，请为您的实时流输入一个短名（无空格）或保留默认值。
11. 对于 WowzaLicenseKey，请输入您在执行获取 Wowza Media Server 3.6 许可证 (p. 200)主题中的步骤时所获得的许可证密钥。如果购买了附加程序，则可以使用竖线 (|) 字符分隔各个密钥值，从而包含更多许可证密钥。
12. 对于 InstanceType，请输入一种实例类型，然后单击 Continue (继续)。默认值为 **`m1.large`**。有关 Amazon EC2 实例类型的更多信息，请参阅可用实例类型。

    您为 InstanceType 输入的值会显示在表中的 API Name (API 名称) 列。实例类型决定着 Wowza Media Server 3.6 实例的定价。有关更多信息，请参阅 Amazon EC2 定价。
13. （可选）在向导的下一页，添加您打算使用的所有标签的键/值对。有关使用标签的更多信息，请参阅向 AWS CloudFormation 堆栈添加标签。单击 Continue (继续)。
14. 检查堆栈的设置。如果您感到满意，请单击 Continue (继续)。然后单击 Close (关闭)。
15. 堆栈创建可能需要花费几分钟时间。要跟踪创建进度，请选择堆栈，然后单击底部框架中的 Events (事件) 选项卡。如果 AWS CloudFormation 无法创建堆栈，则 Events (事件) 选项卡将列出错误消息。

    当您的堆栈准备就绪后，在顶部框架中，堆栈的状态将变为 CREATE_COMPLETE。



16. 当您的堆栈创建完成后，请单击 Outputs (输出) 选项卡，以便查看堆栈创建输出结果。当您在本过程后面部分设置编码器时，您将用到这些值。



下一步：验证 Wowza Media Server 3.6 是否正在运行 (p. 204)

验证 Wowza Media Server 3.6 是否正在运行

在 AWS CloudFormation 创建堆栈后，请执行以下步骤来验证 Wowza Media Server 3.6 是否在您通过 AWS CloudFormation 而配置的 Amazon EC2 实例上运行。

验证 Wowza Media Server 3.6 是否正在运行

1. 登录 AWS 管理控制台，并通过以下网址打开 AWS CloudFormation 控制台：https://console.aws.amazon.com/cloudformation/。
2. 在右上角，单击您在其中创建 AWS CloudFormation 堆栈的区域。
3. 在顶部窗格中，选择您在创建 AWS CloudFormation 实时流堆栈 (p. 201)中创建的堆栈。
4. 在底部窗格中，单击 Outputs (输出) 选项卡。
5. 在 Outputs (输出) 选项卡上，获取 WowzaServerLoginInfo 密钥的值，您将在下一步中将其用于登录证书。
6. 单击 WowzaServerStreamManagerURL 密钥中的 URL，例如，http://ec2-xx-xx-xxx-xxx.compute-1.amazonaws.com:8086/streammanager。当提示您输入登录证书时，请使用您在步骤 5 中获取的 WowzaServerLoginInfo 密钥值。

下一步：设置编码器以发布实时流 (p. 205)

设置编码器以发布实时流

您需要对设备捕获的实时流进行编码，然后才能发送给 Wowza Media Server 3.6。在对实时流进行编码时，您既可以使用适用于 iOS 设备的 Wowza GoCoder 应用，也可以使用支持 RTMP 编码的编码器，如 Telestream Wirecast。

从编码器向 Wowza Media Server 发布流媒体时的步骤因您选择的编码器而异。有关如何配置编码器的更多信息，请转至 Wowza 网站上的特定编码技术或查阅您的编码器文档。

使用以下格式对 Apple HLS 流进行编码：

- 视频：
  - Apple iPhone、iPod 和 iPod touch：H.264 基线档次 3.0 级。当以 iPhone 和 iPod 设备为目标时，请勿使用 B 帧。
  - Apple iPad：H.264 主档次 3.1 级
- 音频：AAC-LC（最高 48 kHz），立体声音频。

平滑流必须同时包含音频和视频。关键帧频率必须介于 1 到 4 秒之间。我们建议关键帧频率为 2 秒。

Wowza Gocoder

要配置 Wowza GoCoder 来发布实时流，请执行以下步骤。

Note

Wowza GoCoder 可从 Apple AppStore 购买。

1. 转至 Wowza 教程如何将 Wowza GoCoder 视频广播 iOS 应用用于 Wowza Media Server。
2. 执行此教程中“配置 Wowza GoCoder 应用”部分的步骤。为 Host (主机) 设置指定以下值：
   a. 通过以下网址打开 AWS CloudFormation 控制台：https://console.aws.amazon.com/cloudformation/。
   b. 在 AWS CloudFormation 控制台的底部窗格中，单击 Outputs (输出) 选项卡。
   c. 复制 WowzaServerDomainName 密钥的值，例如，ec2-***xx-xx-xxx-xxx***.compute-1.amazonaws.com。
   d. 对于 Server (服务器)，请粘贴您刚才复制的值。
   e. 对于 Port (端口)，请输入 1935。
3. 为 Application (应用程序) 设置指定以下值：
   a. 对于 Application (应用程序)，请输入您在创建堆栈时指定的应用程序名称，例如，**`livecf`**。
   b. 在 AWS CloudFormation 控制台的 Outputs (输出) 选项卡中，复制 PublishStreamName 密钥的值，例如，myStream。
   c. 对于 StreamName，请粘贴您在上一步中复制的值。
4. 如果适用，请更改其他值。

RTMP 编码器

RTMP 编码器通常使用以下设置：

发布 URL
: 这是 AWS CloudFormation PublishRTMPURL 密钥的值，例如，rtmp://ec2-*xx-xx-xxx-xxx*.compute-1.amazonaws.com/livecf。

流名称
: 这是 AWS CloudFormation PublishStreamName 密钥的值，例如，myStream。

下一步：使用 Web 应用程序播放实时流 (p. 206)。

使用 Web 应用程序播放实时流

Wowza Media Services 提供了在线示例播放器网页，您可以用来播放来自 Wowza Media Server 分配的实时流。这些播放器可以帮助验证您的流堆栈是否已正确设置。对于其他支持您要使用的流协议的播放器，您可以使用相同的流清单 URL。

执行相应的步骤，以便获得要在您的 Web 页面中为实时流而包含的嵌入代码：

Note

在执行设置编码器以发布实时流 (p. 205)中的相应步骤后，请至少等待 30 秒，然后再通过下面的步骤来播放流内容。

在 Adobe Flash Player 上通过 CloudFront 播放您的实时 HDS 流

1. 通过以下网址打开 AWS CloudFormation 控制台：https://console.aws.amazon.com/cloudformation/。
2. 选择实时 HTTP 流堆栈。
3. 在 AWS CloudFormation 控制台的底部窗格中，单击 Outputs (输出) 选项卡。
4. 复制 PlaybackHDSURL 密钥的值，例如，http://d111111abcdef8.cloudfront.net/livecf/myStream/manifest.f4m。
5. 转至 Wowza 网站上的 Flash HTTP 播放器示例网页，将您在上一步中复制的 URL 粘贴到 Stream (流) 字段，然后单击 Connect (连接)。

在 Apple 或其他设备上通过 CloudFront 播放您的实时 HLS 流

1. 通过以下网址打开 AWS CloudFormation 控制台：https://console.aws.amazon.com/cloudformation/。
2. 选择实时流堆栈。
3. 在 AWS CloudFormation 控制台的底部窗格中，单击 Outputs (输出) 选项卡。
4. 复制 PlaybackHLSURL 密钥的值，例如，http://d111111abcdef8.cloudfront.net/livecf/myStream/playlist.m3u8。
5. 使用以下应用程序之一，转至 Wowza 网站上的 iOS/Mac OS X 示例网页，将您在上一步中复制的 URL 粘贴到 Stream (流) 字段，然后单击 Connect (连接)：
   - 运行 Mac OS X Snow Leopard（10.6 版）或更高版本的计算机上的 Safari Web 浏览器
   - 运行 Mac OS X Snow Leopard（10.6 版）或更高版本的计算机上的 QuickTime Player 10.x 或更高版本
   - Apple iOS 设备上的 Safari Web 浏览器

通过 CloudFront 播放实时平滑流

1. 通过以下网址打开 AWS CloudFormation 控制台：https://console.aws.amazon.com/cloudformation/。
2. 选择实时流堆栈。
3. 在 AWS CloudFormation 控制台的底部窗格中，单击 Outputs (输出) 选项卡。
4. 复制 PlaybackSmoothURL 密钥的值，例如，http://d111111abcdef8.cloudfront.net/livecf/myStream/Manifest。
5. 转至 Wowza 网站上的 Silverlight 播放器示例网页，将您在上一步中复制的 URL 粘贴到 Stream (流) 字段，然后单击 Connect (连接)。

下一步：删除 AWS CloudFormation 实时流堆栈 (p. 207)。

删除 AWS CloudFormation 实时流堆栈

当您的实时事件结束时，请删除为实时流而创建的堆栈。这将删除之前为您的实时流事件而创建的 AWS 资源，并停止资源的按需计费。

删除 AWS CloudFormation 实时流堆栈

1. 通过以下网址打开 AWS CloudFormation 控制台：https://console.aws.amazon.com/cloudformation/。
2. 在右上角，单击您在其中创建堆栈的区域。
3. 选择堆栈，然后单击 Delete Stack (删除堆栈)。
4. 单击 Yes, Delete (是，删除) 以确认。
5. 要跟踪堆栈的删除进度，请选择堆栈，然后单击底部框架中的 Events (事件) 选项卡。

常见问题

使用 CloudFront 和 Wowza Media Server 3.6 的实时 HTTP 流的价格是多少？

使用 CloudFront 和 Wowza Media Server 3.6 的实时 HTTP 流费用包括：

- Wowza Media Server 软件和附加程序：有关更多信息，请参阅 Wowza 网站上的 Licenses Built For You (为您构建的许可证)。
- Amazon EC2：有关更多信息，请参阅按需实例定价表中的 Linux 选项卡。
- CloudFront：有关更多信息，请参阅 Amazon CloudFront 定价。

使用 AWS CloudFormation 不产生任何费用。

我如何使用安全外壳 (SSH) 连接到运行 Wowza Media Server 3.6 的 Amazon EC2 实例？

您只需执行几个步骤，便可以使用 SSH 连接到您的 Amazon EC2 实例。

使用 SSH 连接到运行 Wowza Media Server 3.6 的 Amazon EC2 实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在左侧导航中，单击 Instances (实例)。
3. 右击正确的实例，然后单击 Connect (连接) 以查看关于如何使用 SSH 连接到 Amazon EC2 实例的说明。用户名为 ec2-user。

我如何为我的 Amazon EC2 实例或 CloudFront 分配创建别名记录 (CNAME) 别名？

您的运行 Wowza Media Server 3.6 的 Amazon EC2 实例同时附带一个内部和外部 DNS 名称。Amazon EC2 不提供修改这些 DNS 设置的访问权限。如果要将现有域名映射到您的运行 Wowza Media Server 的 Amazon EC2 实例，请使用 DNS 服务提供商，如 Amazon Route 53。当使用您自己的域名时，我们建议您使用别名记录 (CNAME) 映射到实例的外部 DNS 名称，而不是使用指向实例 IP 地址的 A 记录。

要将您自己的域名映射到 CloudFront 分配，请参阅 使用备用域名（别名记录）(p. 52)。

我是否可以将我的实时事件同时流式传输到兼容 Flash Player 的设备、Apple 设备以及平滑流播放器？

可以，Wowza Media Server 3.6 可以实现同时以 Adobe HTTP 动态流 (Adobe HDS)、Apple HTTP 实时流 (Apple HLS) 和 Microsoft 平滑流格式传送实时流，这些格式分别可以在 Adobe Flash Player 应用程序、Apple iOS 设备和平滑流式播放器上播放。

Wowza Media Server 3.6 是否支持 HTML5？

支持，Wowza Media Server 可以通过以下格式向 HTML5 传送内容：

- 您可以使用 Apple iOS 设备上的 Apple HLS 流格式。
- 您可以使用 Windows 8 设备上的平滑流格式。有关更多信息，请转至 MSDN 网站上的演练：构建您的首个 HTML5 平滑流播放器。
- 对于支持 HTML5 的其他浏览器，您可以使用 Wowza Media Server 通过渐进式下载传送视频。

我是否可以使用 Wowza 和 CloudFront 供应私人实时流？

此时，将无法使用 CloudFront 签名 URL 安全地传送实时流。但是，渐进式下载的媒体可以使用签名 URL 进行私下传送。有关更多信息，请参阅 通过 CloudFront 提供私有内容。(p. 91)。

其他文档

下面的资源可能会在您使用 Wowza 时提供帮助。

Wowza 文档

Wowza 网站包括一些关于 Wowza 实时流与 Amazon Web Services 一起使用的文章、文档和定价信息：

- 适用于 Amazon EC2 的 Wowza 文章
- Wowza Media Server 用户指南
- Amazon Web Services 上的 Wowza

Amazon Web Services 文档

下面的资源包括一些 Amazon Web Services 的用户指南和参考书。

- Amazon Elastic Compute Cloud 文档
- AWS CloudFormation 文档

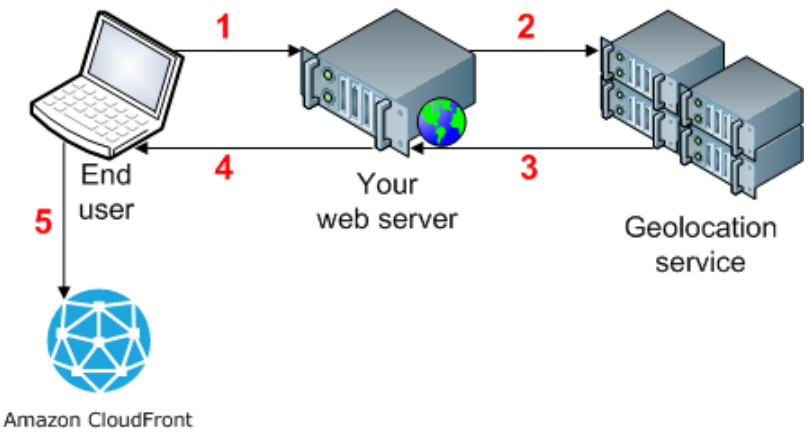
根据地理位置限制访问 CloudFront 分配中的文件（地理阻止）

Topics

Amazon CloudFront 通过将您的 Web 内容（如图像、视频和音频）分发到由遍布全球的节点组成的网络，来提升您网站和应用程序的性能、可靠性和可用性。当最终用户请求您的内容时，CloudFront 将从当时对该用户而言延迟最短的节点向该用户提供您的内容。如果您就可以分发您内容的位置设置地理限制，那么您就可以将 CloudFront 与第三方地理定位服务结合使用，来根据请求的位置控制内容的分发。这称为地理限制或地理阻止。例如，如果请求来自您因版权原因而未被授权将内容分发到的国家/地区，您可阻止该请求并向请求者发出一条消息来解释这种情况。

以下是具体工作原理：

1. 查看您网站的最终用户请求受到地理限制的网页或文件。
2. 您的 Web 应用程序从请求中获得此最终用户的 IP 地址，并将 IP 地址发送给地理定位服务。您将需要有用于这些服务之一的账户。
3. 地理定位服务确定最终用户 IP 地址的地理位置并将结果返回给您的 Web 应用程序。
4. 您的 Web 应用程序将最终用户的位置与文件可（或不可）分发到的位置列表进行比较：
   - 如果允许最终用户访问该网页或文件，您的应用程序将创建 CloudFront 签名 URL 并将其返回给最终用户。
   - 如果不允许最终用户访问该网页或文件，您的 Web 应用程序将“you are not authorized”消息的 URL 返回给最终用户。
5. 如果允许最终用户访问该网页或文件，最终用户的浏览器自动使用签名 URL 从 CloudFront 请求该文件。



通过使用 CloudFront 和第三方地理定位服务限制访问您应用程序层中的内容，您可以全面控制最终用户的体验。对于访问权限被阻止的最终用户，您的应用程序可以显示有明确含义的消息而不是返回错误代码。您还可以根据最终用户的位置自定义为其显示的错误消息。

Note

如果要在按国家/地区疆界划分的地理区域中限制对您内容的分发，或者要限制分发与您的分配有关的所有文件（而不是个别文件），则您可能更喜欢使用 CloudFront 地理限制。有关更多信息，请参阅 限制您的内容的地理分配 (p. 42)。

以下任务列表将引导您完成在应用程序中实现地理阻止功能的过程，以根据最终用户的位置限制访问 CloudFront 分配中的内容。

为根据地址位置限制访问 CloudFront 分配中的文件而需要完成的任务列表

1. 获取用于某项地理定位服务的账户。
   本节提供了 Digital Element 和 MaxMind 的示例代码，但任何地理定位服务都受支持。
2. 如果您还没有 AWS 账户，请创建一个。有关更多信息，请参阅 创建 Amazon Web Services 账户 (p. 211)。
3. 将您的内容上传到 Amazon Simple Storage Service (S3) 存储桶。有关更多信息，请参阅 Amazon S3 文档。
4. 将 Amazon CloudFront 和 Amazon S3 配置为提供私有内容。有关更多信息，请参阅 通过 CloudFront 提供私有内容。 (p. 91)。
5. 编写一个具备以下功能的 Web 应用程序：
   a. 将每个最终用户请求的 IP 地址发送到该地理定位服务。
   b. 评估该地理定位服务的返回值（通常为国家/地区代码），以确定最终用户是否位于您希望 CloudFront 将您的内容分发到的位置。
   c. 为您的 CloudFront 内容生成签名 URL，或阻止访问该内容。

   下面提供了 Digital Element 和 MaxMind 的 Java、.NET 和 PHP 示例代码。请参阅适用主题：

   - Digital Element 的示例代码 (p. 212)
   - MaxMind 的示例代码 (p. 222)

   如果您使用的是其他地理定位服务，请参考它们的相关文档。

Amazon Web Services 提供了适用于 Java、.NET 和 PHP 的开发工具包。有关更多信息，请参阅 Amazon Web Services 网站上的相关页面：

- Java 开发人员中心
- Windows 和 .NET 开发人员中心
- PHP 开发人员中心

创建 Amazon Web Services 账户

Note

AWS 会在您创建账户时自动为该帐户注册所有服务。您只需为使用的服务付费。

如何创建 AWS 账户

1. 请转到 http://aws.amazon.com，然后单击 Create an AWS Account (创建 AWS 账户)。
2. 按照屏幕上的说明进行操作。
   作为注册流程的一部分，您会收到一个电话，需要您使用电话键盘输入一个 PIN 码。

Digital Element 的示例代码

本节中的示例说明了如何根据最终用户的 IP 地址从 Digital Element 中获得相应地理位置。这些示例还说明了在您获允将内容分发到最终用户的位置时如何为所请求的对象创建签名 URL。

所有示例代码在文档发布前都进行了测试，但发布后对 Digital Element API 进行的更改可能会影响这些示例如今的准确性。有关最新信息，请访问 Digital Element 文档。

请参阅相关示例代码：

- Digital Element 的 Java 示例代码 (p. 212)
- Digital Element 的 .NET 示例代码 (p. 216)
- Digital Element 的 PHP 示例代码 (p. 219)

Note

在这些代码示例中，红色斜体文本是占位符。请使用任何适用于您情况的值替换此文本。

Digital Element 的 Java 示例代码

此处提供的代码示例用于获取与最终用户 IP 地址相关的国家/地区代码，并在用户位于允许分发到的地理位置时允许用户访问 CloudFront 内容。就本示例而言，已授权程序将请求的内容分发到澳大利亚（国家/地区代码为 AU）以外的任何国家/地区。

GetCountryCodeServlet.java

GetCountryCodeServlet.java 调用 GetDigitalElementCountryCode.java（在本文后面部分将予以介绍），以向 Digital Element 索取与最终用户 IP 地址相关的国家/地区代码。如果国家/地区代码不是 AU（澳大利亚），GetCountryCodeServlet.java 则调用 SignedUrl.java 以创建一个签名 URL，供最终用户用于访问 CloudFront 分配中的文件。

```
/*
 * 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 向 Digital Element 发送包含最终用户 IP 地址的请求。返回值是一个国家/地区代码。

```
/*
 * 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 创建一个签名 URL 供最终用户用于访问 CloudFront 分配中的文件。

```
/*
 * 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;
   }
}
```

Digital Element 的 .NET 示例代码

以下示例应用程序获取最终用户的 IP 地址并将该 IP 地址发送给 Digital Element。Digital Element 返回与最终用户的 IP 地址对应的国家/地区代码（XML 格式）。该应用程序分析 XML 并评估 Digital Element 返回的值是否与被阻止的国家/地区代码相匹配。如果最终用户的国家/地区被阻止，该应用程序将显示一条消息说明这种情况。如果最终用户的国家/地区未被阻止，该应用程序则创建一个将在一分钟后过期的签名 URL，并执行必要的替换以确保此 URL 不包含任何无效的字符，然后将用户的浏览器重定向至此签名 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>
```

Digital Element 的 PHP 示例代码

以下示例应用程序获取最终用户的 IP 地址并将该 IP 地址发送给 Digital Element。Digital Element 返回与最终用户的 IP 地址对应的国家/地区代码（XML 格式）。然后，该应用程序会分析 XML、显示被阻止的国家代码并评估 Digital Element 返回的值是否与被阻止的国家/地区代码相匹配。如果最终用户所在的国家/地区未被阻止，该应用程序将显示“You are not blocked”消息，使用规范策略创建一个将在五分钟后过期的签名 URL，执行必要的替换以确保此 URL 不包含任何无效的字符，然后将用户的浏览器重定向至此签名 URL。如果最终用户所在的国家/地区被阻止，该应用程序将显示“You are blocked”消息和一张图形。

```
<!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>
```

MaxMind 的示例代码

本节中的示例说明了如何根据最终用户的 IP 地址从 MaxMind 获取相应的地理位置；以及在您被授权将所请求的对象分发到用户所在位置的情况下，如何为该对象创建签名 URL。

所有示例代码在文档发布前都进行了测试，但发布后对 MaxMind API 进行的更改可能会影响示例如今的准确性。有关最新信息，请访问 MaxMind 文档。

请参阅相关示例代码：

- MaxMind 的 Java 示例代码 (p. 222)
- MaxMind 的 .NET 示例代码 (p. 229)
- MaxMind 的 PHP 示例代码 (p. 226)

MaxMind 的 Java 示例代码

GetCountryCodeServlet.java

GetCountryCodeServlet.java 调用 GetMaxMindCountryCode.java（在本文后面部分将予以介绍），以向 MaxMind 索取与最终用户 IP 地址相关的国家/地区代码。如果国家/地区代码不是 AU（澳大利亚），GetCountryCodeServlet.java 则调用 SignedUrl.java 以创建一个签名 URL，供最终用户用于访问 CloudFront 分配中的文件。

```
/*
 * 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 向 MaxMind 发送包含最终用户 IP 地址的请求。返回值是一个国家/地区代码。

```
/*
 * 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 创建一个签名 URL 供最终用户用于访问 CloudFront 分配中的文件。

```
/*
 * 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;
   }
}
```

MaxMind 的 PHP 示例代码

以下示例应用程序获取最终用户的 IP 地址并将该 IP 地址发送给 MaxMind。MaxMind 返回与最终用户的 IP 地址对应的国家/地区代码。然后，该应用程序显示被阻止的国家/地区代码并评估 MaxMind 返回的值是否与被阻止的国家/地区代码相匹配。如果最终用户所在的国家/地区未被阻止，该应用程序将显示“You are not blocked”消息，使用规范策略创建一个将在五分钟后过期的签名 URL，执行必要的替换以确保此 URL 不包含任何无效的字符，然后将用户的浏览器重定向至此签名 URL。如果最终用户所在的国家/地区被阻止，该应用程序将显示“You are blocked”消息和一张图形。

```
<!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>
```

MaxMind 的 .NET 示例代码

以下示例应用程序获取最终用户的 IP 地址并将该 IP 地址发送给 MaxMind。MaxMind 返回与最终用户的 IP 地址对应的国家/地区代码。然后，该应用程序评估 Digital Element 返回的值是否与被阻止的国家/地区代码相匹配。如果最终用户的国家/地区被阻止，该应用程序将显示一条消息说明这种情况。如果最终用户的国家/地区未被阻止，该应用程序则创建一个将在一分钟后过期的签名 URL，并执行必要的替换以确保此 URL 不包含任何无效的字符，然后将用户的浏览器重定向至此签名 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>
```

常见问题

我如何才能确保获取访问我网站的最终用户的正确 IP 地址？

您可使用多种方法来获取访问您网站的最终用户的 IP 地址。下面是两种可以采用的方法：

- 如果您的 Web 服务器未通过负载均衡器连接到互联网，您可使用 Web 服务器变量来获取远程 IP 地址。但是，此 IP 地址并不一定是最终用户的 IP 地址 — 它也可能是代理服务器的 IP 地址，具体取决于最终用户如何连接到互联网。
- 如果您的 Web 服务器通过负载均衡器连接到互联网，Web 服务器变量可能包含负载均衡器的 IP 地址，而不是最终用户的 IP 地址。在这种配置中，我们建议您使用 `X-Forwarded-For` http 标头中的最后一个 IP 地址。该标头通常包含多个 IP 地址，其中大部分是代理或负载均衡器的 IP 地址。列表中最后一个 IP 地址是最有可能与最终用户地理位置有关的。

如果您的 Web 服务器未连接至负载均衡器，我们建议您使用 Web 服务器变量，而不是 `X-Forwarded-For` 标头，以避免遭到 IP 地址欺骗攻击。本文档中的示例代码在 `X-Forwarded-For` 标头存在的情况下会使用该标头。如果您不想使用该方法来获取最终用户的 IP 地址，您可编辑该示例代码。

我是否可以使用任何第三方地理定位服务来限制访问 CloudFront 中的内容？

可以。您将需要使用第三方服务的账户来调用它们的 API，并且您还需要相应地修改示例代码。

使用该解决方案的费用是多少？

使用第三方地理定位服务的费用将取决于您使用哪家服务提供商。Amazon CloudFront 定价页面上提供了目前的 CloudFront 使用价格。使用 CloudFront 私有内容功能不需支付任何额外的 CloudFront 费用。

我能否使用国家/地区以外的位置信息来阻止对我内容的访问？

如果您的地理定位服务提供国家/地区代码之外的信息，您的应用程序可使用此信息来确定您是否可将您的内容分发给最终用户。然后，您的应用程序便可生成 CloudFront 签名 URL；有关说明，请参阅本教程或 Amazon CloudFront 开发人员指南中的使用签名 URL 提供私有内容。

如果第三方服务未返回有关最终用户的正确信息，我该怎么办？

确认您是否正确调用了第三方地理定位服务提供的 API 以及您使用了正确的最终用户 IP 地址。如果您仍遇到与第三方服务有关的问题，或与从该服务接收的数据的准确性有关的问题，请直接联系服务供应商。

其他服务和文档

Digital Element 服务和文档

有关 Digital Element 服务的信息，请参阅 Digital Element 网站。

Digital Element 服务文档仅可通过 Digital Element 账户获得。

MaxMind 服务和文档

MaxMind 提供各种地理定位服务和其他 Web 服务，其中包括以下服务：

- MaxMind GeoIP Omni Web 服务，网址为 http://www.maxmind.com/app/web_services_omni
- MaxMind JavaScript Web 服务，网址为 http://www.maxmind.com/app/javascript
- 其他 MaxMind Web 服务，网址为 http://www.maxmind.com/app/web_services

每个 MaxMind API 的 Web 分配均包含文档和示例程序。

有关更多信息，请参阅将 MaxMind 与 Amazon CloudFront 配合使用。

Amazon Web Services 文档

- CloudFront，网址为 http://aws.amazon.com/documentation/cloudfront
- Amazon S3，网址为 http://aws.amazon.com/documentation/s3/

使用 CloudFront 和 Adobe Flash Player 的按需视频流

当您使用 CloudFront 来流式传输媒体文件时，您需要同时提供您的媒体文件以及您希望最终用户用来播放媒体文件的媒体播放器。要使用 Adobe Flash Player 来流式传输采用 CloudFront 的媒体文件，请执行以下主题中的步骤：

1. 创建 Amazon S3 存储桶 (p. 234)
2. 创建 CloudFront Web 和 RTMP 分配 (p. 234)
3. 使用 Adobe Flash Builder 创建 Flash 项目 (p. 235)
4. 将媒体和 Flash Builder 文件上传到 Amazon S3 存储桶 (p. 237)
5. 播放媒体文件 (p. 238)

本教程使用 Adobe Flash Builder 4.6 版本来产生必需的文件，以便使用 Adobe Flash Player 流式传输视频。有关 Flash Builder 的更多信息，请转至 Adobe 网站上的 Flash Builder 页面。有关下载 Adobe Flash Builder 免费试用版本的信息，请转至下载/Adobe Flash Builder 4.6 页面。

有关 Flash Player 支持的编解码器列表，请转至 Adobe 网站上的支持的编解码器 | Flash Player。

有关使用 CloudFront 来流式传输媒体内容的更多信息，请参阅使用 RTMP 分配 (p. 43)。

创建 Amazon S3 存储桶

您可将媒体文件和媒体播放器文件上传到同一 Amazon S3 存储桶或不同的存储桶。在本教程中，您将为媒体文件和 Flash Player 媒体播放器文件创建同一存储桶。在本过程的后面部分中，您将在创建 Adobe Flash Player 文件后上传这些文件。

创建 Amazon S3 存储桶

1. 登录 AWS 管理控制台，并通过以下网址打开 Amazon S3 控制台：https://console.aws.amazon.com/s3/。
2. 在 Amazon S3 控制台中，单击 Create Bucket (创建存储桶)。
3. 在 Create Bucket (创建存储桶) 对话框中，输入存储桶名称。

   Important

   要使您的存储桶能够与 CloudFront 一起使用，其名称必须符合 DNS 命名要求。有关详细信息，请参阅 *Amazon Simple Storage Service 开发者指南* 中的 Bucket Restrictions and Limitations。

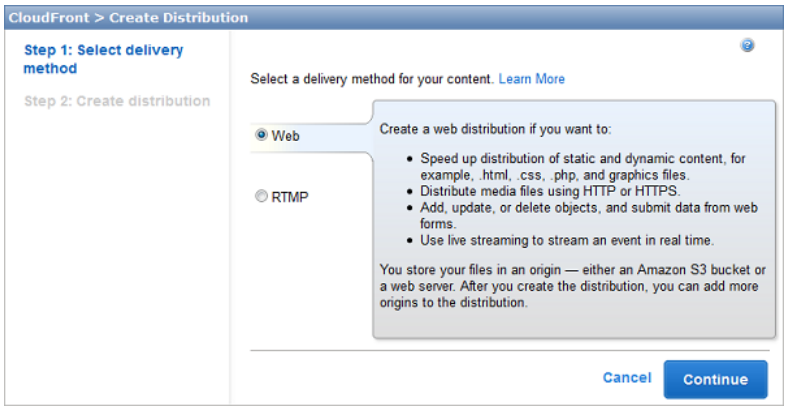
4. 选择您的存储桶区域。默认情况下，Amazon S3 在美国标准地区中创建存储桶。我们建议您选择一个靠近您的地区，以便优化延迟，尽可能降低成本或满足法规要求。
5. 单击 Create (创建)。

创建 CloudFront Web 和 RTMP 分配

要配置 CloudFront 以便流式传输媒体文件，您需要一个 CloudFront RTMP 分配。在本教程中，您还将创建一个 CloudFront Web 分配，以便访问 Adobe Flash Builder 所创建的 .hlml 文件。执行以下两个步骤。

创建 CloudFront Web 分配

1. 通过以下网址打开 Amazon CloudFront 控制台：https://console.aws.amazon.com/cloudfront/。
2. 单击 Create Distribution (创建分配)。
3. 在 Create Distribution Wizard (创建分配向导) 的第一页上，接受默认选择 Web，然后单击 Continue (继续)。



4. 在向导的第二页，单击 Origin Domain Name (源域名) 字段，然后选择您在步骤 创建 Amazon S3 存储桶 (p. 234) 中创建的 Amazon S3 存储桶。如果您有很多 Amazon S3 存储桶，则可以键入存储桶名称的前几个字符，以便在列表中筛选。
5. 接受其余字段的默认值，然后单击 Create Distribution (创建分配)。
6. 在 CloudFront 创建分配后，分配的 Status (状态) 列的值将从 InProgress (进行中) 改为 Deployed (已部署)。这应该需要不到 15 分钟的时间。

   CloudFront 指派给您的分配的域名将出现在分配列表中。（它同时也出现在选定分配的 General (常规) 选项卡上。）

创建 CloudFront RTMP 分配

1. 在 CloudFront 控制台中，单击 Create Distribution (创建分配)。
2. 在 Create Distribution Wizard (创建分配向导) 中，单击 RTMP，然后单击 Continue (继续)。
3. 在向导的第二页，单击 Origin Domain Name (源域名) 字段，然后选择您在步骤 创建 Amazon S3 存储桶 (p. 234) 中创建的 Amazon S3 存储桶。如果您有很多 Amazon S3 存储桶，则可以键入存储桶名称的前几个字符，以便在列表中筛选。
4. 接受 Create Distribution (创建分配) 页面上其余字段的默认值，然后单击 Create Distribution (创建分配)。
5. 在 CloudFront 创建分配后，分配的 Status (状态) 列的值将从 InProgress (进行中) 改为 Deployed (已部署)。这应该需要不到 15 分钟的时间。

   CloudFront 指派给您的分配的域名将出现在分配列表中。域名同时也出现在选定分配的 General (常规) 选项卡上。

使用 Adobe Flash Builder 创建 Flash 项目

您可使用 Adobe Flash Builder 自动创建 Flash 项目，其中包含使用 Adobe Flash 播放媒体文件所需的全部文件。

使用 Adobe Flash Builder 创建 Flash 项目

1. 启动 Adobe Flash Builder。
2. 在 Flash Builder File (文件) 菜单上，单击 New (新建) > Flex Project (Flex 项目)。
3. 输入以下值：

   - Project name: (项目名称:)为您的项目输入一个名称，例如，CloudFrontStreaming。
   - Folder: (文件夹:)指定您希望 Flash Builder 用来保存此项目文件的位置。如果您不想使用默认位置，请取消选中 Use default location (使用默认位置) 复选框，然后选择其他位置。

     记录该位置；您将在本过程后面部分中用到它。
   - Application type: (应用程序类型:)接受默认值 Web。
   - Flex SDK version: (Flex 开发工具包版本:)接受默认值 Use default SDK (使用默认 SDK)。

4. 要创建项目，请单击 Finish (完成)。

   在 Flash Builder 创建项目之后，具有项目名称的新选项卡将出现在 Flash Builder 用户界面中。<project-name> 选项卡上的 Source (源) 按钮处于选中状态，并且 Source (源) 页面包含多行 XML 代码。
5. 删除 Source (源) 页面上的默认 XML 代码。
6. 复制下面的 XML 代码，并将其粘贴到 Adobe Flash Builder 中的空白 Source (源) 页面。

```
<?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. 在粘贴到 Source (源) 页面的 XML 代码中，请替换以下值：

   - 将 RTMP-DISTRIBUTION-DOMAIN-NAME 替换为您的 RTMP 分配的 CloudFront 域名，例如，s5c39gqb8ow64r.cloudfront.net。

- 将 `VIDEO-FILE-NAME-WITHOUT-EXTENSION` 替换为您的视频文件，但不包括文件扩展名。例如，如果您的视频名称为 my-vacation.mp4，则只需输入 `my-vacation`。

8. 保存您的更改。
9. 在 Flash Builder 的 Project (项目) 菜单上，单击 Export Release Build (导出发布版本)。
10. 在 Export Release Build (导出发布版本) 对话框中，接受所有默认值，然后单击 Finish (完成)。

    Flash Builder 会为您的项目创建文件，并将其保存到您在步骤 3 中指定的位置。

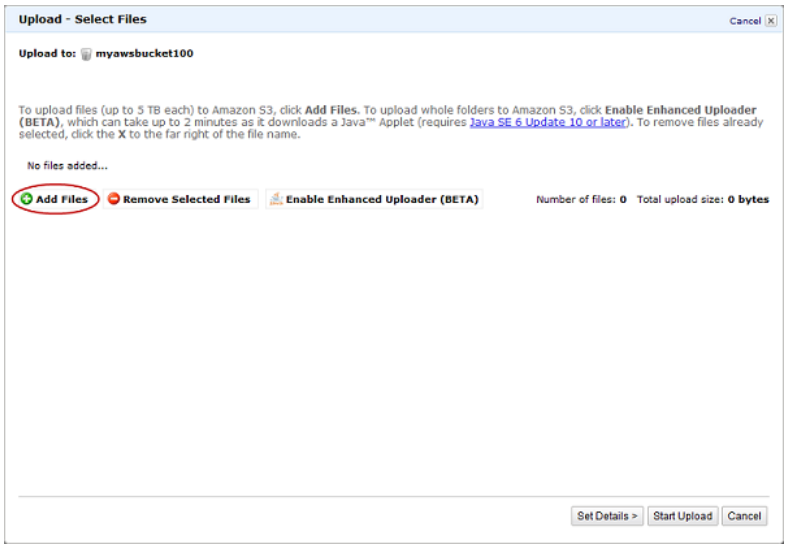
将媒体和 Flash Builder 文件上传到 Amazon S3 存储桶

当您使用 Adobe Flash Builder 生成用于流式传输的媒体文件时，您需要将媒体文件和 Flash Builder 文件上传到同一 Amazon S3 存储桶。

将媒体文件和 Flash Builder 文件上传到 Amazon S3 存储桶

1. 登录 AWS 管理控制台，并通过以下网址打开 Amazon S3 控制台：https://console.aws.amazon.com/s3/。
2. 在 Buckets (存储桶) 窗格中，选择您的存储桶，然后单击 Upload (上传)。
3. 在 Upload - Select Files (上传 - 选择文件) 页面上，单击 Add Files (添加文件)，然后添加以下文件：

   - 您的媒体文件
   - Flash Builder 在您执行步骤 使用 Adobe Flash Builder 创建 Flash 项目 (p. 236) 时所生成的文件。仅上传 `bin-release` 目录中的文件。您可排除 `bin-release/history` 子目录中的文件。



4. 授予您在上一步中所添加文件的公共读取权限。

   a. 单击 Set Details (设置详细信息)。
   b. 在 Set Details (设置详细信息) 页面上，单击 Set Permissions (设置权限)。
   c. 在 Set Permissions (设置权限) 页面上，单击 Make everything public (公开一切信息)。

5. 单击 Start Upload (开始上传)。

播放媒体文件

要播放媒体文件，您需要显示 Flash Builder 为您的项目而创建的 HTML 文件以及您上传到 Amazon S3 存储桶的 HTML 文件。

播放媒体文件

1. 通过串联下面的值，输入 Flash Builder 为您的项目而创建的 HTML 文件的 CloudFront URL：

   `http://`*`domain-name-for-your-CloudFront-distribution`*`/`*`HTML-file-name`*

   例如：

   `http://d111111abcdef8.cloudfront.net/CloudFrontStreaming.html`
2. 在视频播放器中，单击箭头按钮。

   视频应该会开始播放。

使用 CloudFront 和 Flowplayer for Adobe Flash 的按需视频流

当您使用 CloudFront 来流式传输媒体文件时，您需要同时提供您的媒体文件以及您希望最终用户用来播放媒体文件的媒体播放器。要使用 Flowplayer for Adobe Flash 媒体播放器来流式传输采用 CloudFront 的媒体文件，请执行以下主题中的步骤：

1. 将媒体和 Flowplayer 文件上传到 Amazon S3 存储桶 (p. 239)
2. 创建 CloudFront Web 和 RTMP 分配 (p. 240)
3. 在 HTML 页面中嵌入视频 (p. 241)

Note

要使用 CloudFront 和 Flowplayer for Adobe Flash 来流式传输视频，用户必须在其浏览器中启用 Javascript。

本教程基于 Flowplayer for Adobe Flash 3.2.12 版。有关 Flowplayer Flash 的更多信息，请转至 Flowplayer Flash 网站。有关 Flowplayer Flash 支持的视频格式列表，请转至与 Flowplayer 开发环境相关的 Flowplayer 开发人员文档中的视频格式部分。

Note

Flowplayer 已经发布了其媒体播放器的 HTML 5 版本。以下步骤仅用于 Flowplayer Flash，而不适用于 Flowplayer HTML5。

有关使用 CloudFront 来流式传输媒体内容的更多信息，请参阅使用 RTMP 分配 (p. 43)。

将媒体和 Flowplayer 文件上传到 Amazon S3 存储桶

您可将媒体文件和媒体播放器文件上传到同一 Amazon S3 存储桶或不同的存储桶。在本教程中，您将把 .mp4 媒体文件和 Flowplayer 媒体播放器文件上传到同一存储桶中。

将媒体文件和 Flowplayer 文件上传到同一 Amazon S3 存储桶

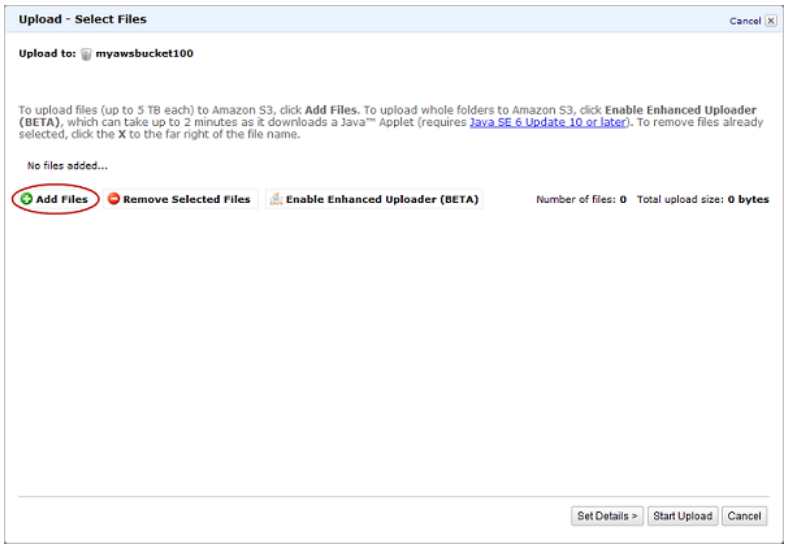
1. 从 Flowplayer 网站下载以下文件：
   - Flowplayer 媒体播放器。在您下载 Flowplayer 后，请解压缩 .zip 文件的内容。
   - flowplayer.rtmp-3.2.10.swf。此插件可以让 Flowplayer 使用 RTMP 协议流式传输视频。此文件位于 Flowplayer 网站的 RTMP 页面上。
2. 登录 AWS 管理控制台，并通过以下网址打开 Amazon S3 控制台：https://console.aws.amazon.com/s3/。
3. 在 Amazon S3 控制台中，单击 Create Bucket (创建存储桶)。
4. 在 Create Bucket (创建存储桶) 对话框中，输入存储桶名称。

Important

要使您的存储桶能够与 CloudFront 一起使用，其名称必须符合 DNS 命名要求。有关详细信息，请参阅 *Amazon Simple Storage Service 开发者指南* 中的 Bucket Restrictions and Limitations。

5. 选择您的存储桶区域。默认情况下，Amazon S3 在美国标准地区中创建存储桶。我们建议您选择一个靠近您的地区，以便优化延迟，尽可能降低成本或满足法规要求。
6. 单击 Create (创建)。
7. 在 Buckets (存储桶) 窗格中选择您的存储桶，然后单击 Upload (上传)。
8. 在 Upload - Select Files (上传 - 选择文件) 页面上，单击 Add Files (添加文件)，然后添加以下文件（您文件中的 Flowplayer 版本号可能有所不同）：

   - flowplayer.controls-3.2.12.swf
   - flowplayer-3.2.11.min.js
   - flowplayer-3.2.12.swf
   - flowplayer.rtmp-3.2.10.swf
   - 您的 .mp4 格式媒体文件



9. 授予您在上一步中所添加文件的公共读取权限。

   a. 单击 Set Details (设置详细信息)。
   b. 在 Set Details (设置详细信息) 页面上，单击 Set Permissions (设置权限)。
   c. 在 Set Permissions (设置权限) 页面上，单击 Make everything public (公开一切信息)。

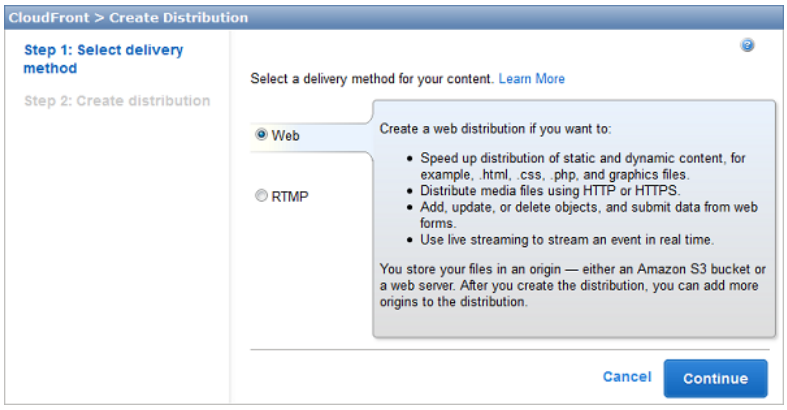
10. 单击 Start Upload (开始上传)。

创建 CloudFront Web 和 RTMP 分配

要配置 CloudFront 来流式传输媒体文件，您需要一个用于 Flowplayer 文件的 CloudFront Web 分配以及一个用于媒体文件的 RTMP 分配。执行以下两个步骤可创建 Web 分配和 RTMP 分配。

为您的 Flowplayer 文件创建 CloudFront Web 分配

1. 通过以下网址打开 Amazon CloudFront 控制台：https://console.aws.amazon.com/cloudfront/。
2. 单击 Create Distribution (创建分配)。
3. 在 Create Distribution Wizard (创建分配向导) 的第一页上，接受默认选择 Web，然后单击 Continue (继续)。



4. 在向导的第二页，单击 Origin Domain Name (源域名) 字段，然后选择您在步骤 将媒体文件和 Flowplayer 文件上传到同一 Amazon S3 存储桶 (p. 239) 中创建的 Amazon S3 存储桶。如果您有很多 Amazon S3 存储桶，则可以键入存储桶名称的前几个字符，以便在列表中筛选。
5. 接受其余字段的默认值，然后单击 Create Distribution (创建分配)。
6. 在 CloudFront 创建分配后，分配的 Status (状态) 列的值将从 InProgress (进行中) 改为 Deployed (已部署)。这应该需要不到 15 分钟的时间。

   CloudFront 指派给您的分配的域名将出现在分配列表中。（它同时也出现在选定分配的 General (常规) 选项卡上。）

为您的媒体文件创建 CloudFront RTMP 分配

1. 在 CloudFront 控制台中，单击 Create Distribution (创建分配)。
2. 在 Create Distribution Wizard (创建分配向导) 中，单击 RTMP，然后单击 Continue (继续)。
3. 在向导的第二页，单击 Origin Domain Name (源域名) 字段，然后选择您在步骤 将媒体文件和 Flowplayer 文件上传到同一 Amazon S3 存储桶 (p. 239) 中创建的 Amazon S3 存储桶。如果您有很多 Amazon S3 存储桶，则可以键入存储桶名称的前几个字符，以便在列表中筛选。
4. 接受 Create Distribution (创建分配) 页面上其余字段的默认值，然后单击 Create Distribution (创建分配)。
5. 在 CloudFront 创建分配后，分配的 Status (状态) 列的值将从 InProgress (进行中) 改为 Deployed (已部署)。这应该需要不到 15 分钟的时间。

   CloudFront 指派给您的分配的域名将出现在分配列表中。域名同时也出现在选定分配的 General (常规) 选项卡上。

在 HTML 页面中嵌入视频

以下示例 HTML 文件展示了如何使用您在创建 CloudFront Web 和 RTMP 分配 (p. 240)中创建的 Web 和 RTMP 分配来流式传输视频。要使用此示例来流式传输视频，请执行以下步骤：

1. 复制下列 HTML 代码，然后将其粘贴到文本编辑器中。
2. 查看 HTML 文件中的注释，并将以下占位符替换为适用的值：

   - WEB-DISTRIBUTION-DOMAIN-NAME
   - VIDEO-FILE-NAME
   - RTMP-DISTRIBUTION-DOMAIN-NAME

3. 使用 .html 文件扩展名保存文件，例如，`flowplayer-example.html`。

4. 在 Web 浏览器中打开 .html 文件，并播放您的视频。

```
<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>
```

使用 CloudFront 和 JW Player 的按需视频流

当您使用 CloudFront 来流式传输媒体文件时，您需要同时提供您的媒体文件以及您希望最终用户用来播放媒体文件的媒体播放器。要使用 JW Player 媒体播放器来流式传输采用 CloudFront 的媒体文件，请执行以下主题中的步骤：

1. 将媒体文件和 JW Player 文件上传到 Amazon S3 存储桶 (p. 244)
2. 创建 CloudFront Web 和 RTMP 分配 (p. 245)
3. 在网页中嵌入视频 (p. 246)
4. 上传 HTML 文件并播放视频 (p. 247)

本教程基于 JW Player 6.1 免费版。有关 JW Player 的更多信息，请转至 JW Player 网站。有关 JW Player 支持的视频格式列表，请转至 JW Player 功能页面。

有关使用 CloudFront 来流式传输媒体内容的更多信息，请参阅使用 RTMP 分配 (p. 43)。

将媒体文件和 JW Player 文件上传到 Amazon S3 存储桶

您可将媒体文件和媒体播放器文件上传到同一 Amazon S3 存储桶或不同的存储桶。在本教程中，您将把 .mp4 或 .flv 媒体文件和 JW Player 媒体播放器文件上传到同一存储桶中。

将媒体文件和 JW Player 文件上传到同一 Amazon S3 存储桶

1. 如果您还没有 JW Player 媒体播放器文件，请从 JW Player 网站上的功能页面下载该播放器。然后解压缩 .zip 文件的内容。
2. 登录 AWS 管理控制台，并通过以下网址打开 Amazon S3 控制台：https://console.aws.amazon.com/s3/。
3. 在 Amazon S3 控制台中，单击 Create Bucket (创建存储桶)。
4. 在 Create Bucket (创建存储桶) 对话框中，输入存储桶名称。

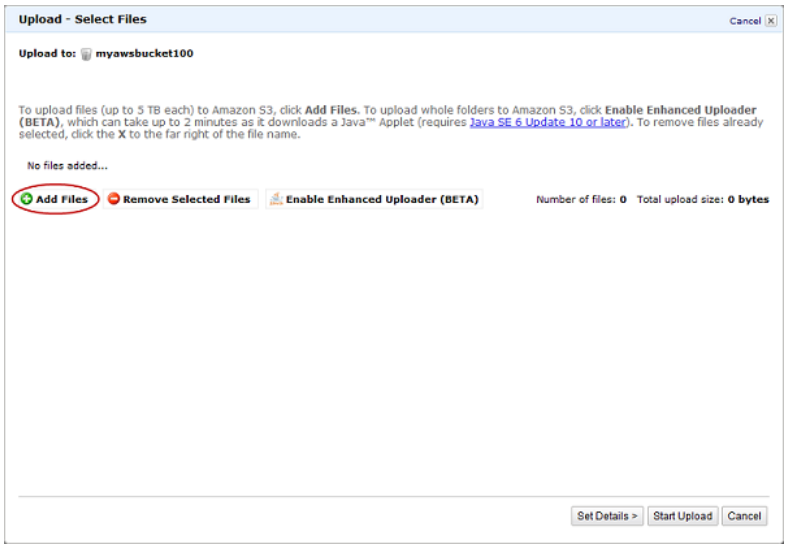
   

   Important

   要使您的存储桶能够与 CloudFront 一起使用，其名称必须符合 DNS 命名要求。有关详细信息，请参阅 *Amazon Simple Storage Service 开发者指南* 中的 Bucket Restrictions and Limitations。

5. 选择您的存储桶区域。默认情况下，Amazon S3 在美国标准地区中创建存储桶。我们建议您选择一个靠近您的地区，以便优化延迟，尽可能降低成本或满足法规要求。
6. 单击 Create (创建)。
7. 在 Buckets (存储桶) 窗格中选择您的存储桶，然后单击 Upload (上传)。
8. 在 Upload - Select Files (上传 - 选择文件) 页面上，单击 Add Files (添加文件)，然后添加以下文件：

   - jwplayer.flash.swf
   - jwplayer.html5.js
   - jwplayer.js
   - 您的 .mp4 或 .flv 媒体文件。



9. 授予您在上一步中所添加文件的公共读取权限。

   a. 单击 Set Details (设置详细信息)。
   b. 在 Set Details (设置详细信息) 页面上，单击 Set Permissions (设置权限)。
   c. 在 Set Permissions (设置权限) 页面上，单击 Make everything public (公开一切信息)。

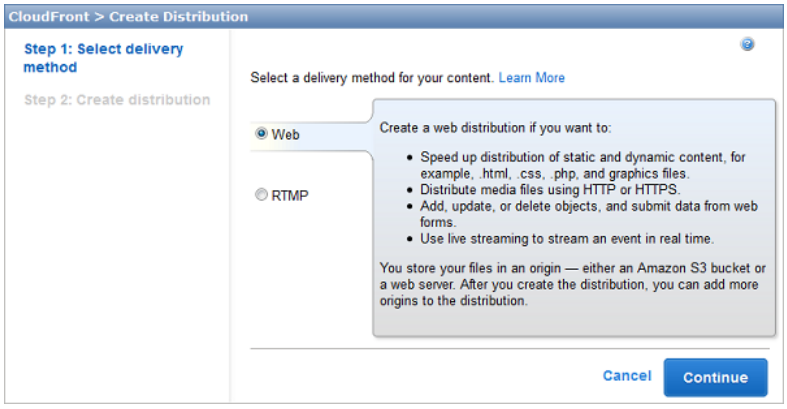
10. 单击 Start Upload (开始上传)。

创建 CloudFront Web 和 RTMP 分配

要配置 CloudFront 来流式传输媒体文件，您需要一个用于 JW Player 文件的 CloudFront Web 分配、一个 HTML 文件以及一个用于媒体文件的 RTMP 分配。执行以下两个步骤可创建 Web 分配和 RTMP 分配。

为您的 JW Player 文件创建 CloudFront Web 分配

1. 通过以下网址打开 Amazon CloudFront 控制台：https://console.aws.amazon.com/cloudfront/。
2. 单击 Create Distribution (创建分配)。
3. 在 Create Distribution Wizard (创建分配向导) 的第一页上，接受默认选择 Web，然后单击 Continue (继续)。



4. 在向导的第二页，单击 Origin Domain Name (源域名) 字段，然后选择您在步骤 将媒体文件和 JW Player 文件上传到同一 Amazon S3 存储桶 (p. 244) 中创建的 Amazon S3 存储桶。如果您有很多 Amazon S3 存储桶，则可以键入存储桶名称的前几个字符，以便在列表中筛选。
5. 接受其余字段的默认值，然后单击 Create Distribution (创建分配)。
6. 在 CloudFront 创建分配后，分配的 Status (状态) 列的值将从 InProgress (进行中) 改为 Deployed (已部署)。这应该需要不到 15 分钟的时间。

   CloudFront 指派给您的分配的域名将出现在分配列表中。域名同时也出现在选定分配的“Distribution Settings (分配设置)”页面上。

为您的媒体文件创建 CloudFront RTMP 分配

1. 在 CloudFront 控制台中，单击 Create Distribution (创建分配)。
2. 在 Create Distribution Wizard (创建分配向导) 中，单击 RTMP，然后单击 Continue (继续)。
3. 在向导的第二页，单击 Origin Domain Name (源域名) 字段，然后选择您在步骤 将媒体文件和 JW Player 文件上传到同一 Amazon S3 存储桶 (p. 244) 中创建的 Amazon S3 存储桶。如果您有很多 Amazon S3 存储桶，则可以键入存储桶名称的前几个字符，以便在列表中筛选。
4. 接受 Create Distribution (创建分配) 页面上其余字段的默认值，然后单击 Create Distribution (创建分配)。
5. 在 CloudFront 创建分配后，分配的 Status (状态) 列的值将从 InProgress (进行中) 改为 Deployed (已部署)。这应该需要不到 15 分钟的时间。

   CloudFront 指派给您的分配的域名将出现在分配列表中。域名同时也出现在选定分配的“Distribution Settings (分配设置)”页面上。

在网页中嵌入视频

下面的示例展示了如何使用您在创建 CloudFront Web 和 RTMP 分配 (p. 245)中创建的 Web 和 RTMP 分配在网页中嵌入视频。

Note

您可以使用 JW Player 设置向导获取添加到 HTML 文件的代码。有关更多信息，请参阅 JW Player 网站上的设置向导页面。

执行以下步骤：

1. 复制下列 HTML 代码，然后将其粘贴到文本编辑器中。
2. 查看 HTML 文件中的注释，并将以下占位符替换为适用的值：

   - WEB-DISTRIBUTION-DOMAIN-NAME
   - RTMP-DISTRIBUTION-DOMAIN-NAME
   - VIDEO-FILE-NAME

3. 使用 .html 文件扩展名保存文件，例如，`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>
```

上传 HTML 文件并播放视频

要使用您在在网页中嵌入视频 (p. 246)中创建的 HTML 文件来播放视频，请将文件上传到您的 Amazon S3 存储桶，然后使用您的 CloudFront 分配的 URL。

上传 HTML 文件并播放视频

1. 通过以下网址打开 Amazon S3 控制台：https://console.aws.amazon.com/s3/。
2. 选择您的存储桶，然后单击 Upload (上传)。
3. 在 Upload - Select Files (上传 - 选择文件) 页面上，单击 Add Files (添加文件)，然后添加您的 HTML 文件。
4. 授予您在上一步中添加的 HTML 文件的公共读取权限。

   a. 单击 Set Details (设置详细信息)。
   b. 在 Set Details (设置详细信息) 页面上，单击 Set Permissions (设置权限)。
   c. 在 Set Permissions (设置权限) 页面上，单击 Make everything public (公开一切信息)。

5. 单击 Start Upload (开始上传)。
6. 要播放视频，请在 Web 浏览器中输入以下 URL：

   `http://domain name of your CloudFront distribution/your HTML file name`

Amazon CloudFront 资源

下列相关资源在您使用此服务的过程中会有所帮助。

- Amazon CloudFront API 参考 – 提供了架构位置；完整说明了 API 操作、参数和数据类型；并提供了该服务返回的错误列表。
- Amazon CloudFront 发行说明 – 简要概述了当前版本，并重点介绍了新增功能、修正和已知问题。
- Amazon Simple Storage Service (S3) 技术文档 – 详细探讨了 Amazon S3 服务，包括入门基础知识、服务概述、编程参考和 API 参考。
- Amazon CloudFront 产品信息 – 提供 Amazon CloudFront 相关信息（包括功能和定价信息）的主要网页。
- 开发论坛 – 一种社区形式的论坛，供开发人员讨论与 Amazon CloudFront 有关的技术问题。

- AWS 开发人员工具 – 指向开发人员工具和资源的链接，其中提供了文档、代码示例、发行说明和有助于您利用 AWS 构建创新应用程序的其他信息。
- AWS Support 中心 – AWS 技术支持的主页，可通过它访问我们的开发人员论坛、技术常见问题、服务状态页面，以及 AWS Premium Support（如果您已订阅此计划）。
- AWS Premium Support 信息 – 提供有关 AWS Premium Support 信息的主要 Web 页面，它是一种一对一、快速响应的支持渠道，可帮助您在 AWS 基础设施服务上构建和运行应用程序。
- 联系我们 – 用于咨询有关您的账单或账户的问题的链接。如有技术问题，请使用上述开发论坛或支持连接。
- 使用条款 – 有关我们的版权和商标的详细信息；您的账户、许可、网站访问和其他主题。

我从这里可以继续进行哪些内容？

CloudFront尽管使用起来相当简单，但功能也十分丰富。有各种各样的资源可供您用来了解有关 CloudFront 的更多信息。除 Amazon CloudFront 文档外，相关资源还包括 AWS 开发工具包、功能摘要以及与 CloudFront 搭配使用的第三方应用程序的链接。以下列表包含了其中一些可能对您有帮助的资源。

Topics

Amazon CloudFront 开发工具包

AWS 提供了可帮助您以编程方式访问 CloudFront 的开发工具包。

- 适用于 Java 的 AWS 开发工具包
- 适用于 .NET 的 AWS 开发工具包
- 适用于 PHP 的 AWS 开发工具包
- 适用于 Ruby 的 AWS 开发工具包

使用 CloudFront 日志记录

以下 AWS 博客文章讨论了 CloudFront 日志记录功能的增强，以及分析访问日志的一些方式。

AWS 博客： Amazon CloudFront 请求日志记录（针对通过 HTTP 提供的内容）

AWS 博客： Amazon CloudFront 现在支持对访问日志进行流式处理（针对通过 RTMP 提供的内容）

AWS 博客： 增强的 CloudFront 日志，现在包含查询字符串

设置默认根对象

CloudBerry 实验室：如何使用 CloudBerry S3 浏览器设置 CloudFront 默认对象

使对象失效

除 CloudFront 提供的失效方法之外，您还可使用以下第三方工具使对象失效。

Note

这些工具是由与 Amazon Web Services 无关的第三方供应商开发的。有关如何使用这些工具的信息，请参考相应供应商的文档或联系相应供应商。

- CloudBuddy Personal，网址为 http://m1.mycloudbuddy.com/index.html
- CloudBerry Explorer，网址为 http://cloudberrylab.com
- Ylastic，网址为 http://ylastic.com
- Cyberduck，网址为 http://cyberduck.ch
- Bucket Explorer，网址为 http://www.bucketexplorer.com
- CloudFront Invalidator，网址为 http://www.swook.net/p/cloudfront-invalidator.html
- CDN Planet CloudFront Purge Tool，网址为 http://www.cdnplanet.com/tools/cloudfront-purge-tool/

您也可在 Github 上搜索代码示例，网址为：https://github.com。请搜索“CloudFront invalidation”词组。

分配流媒体

StreamingMedia.com：如何开始使用 Amazon CloudFront 流式处理

loncannon.net：

- 使用 Amazon S3 和 CloudFront 的 iPhone 窗口化 HTTP 实时流式处理概念验证
- HTTP 实时视频流分段和分配器
- iPhone 窗口化 HTTP 实时流式处理服务器

Flowplayer.org：带宽检测：确保向所有观众呈现精良品质

JW Player：使用 RTMP 流式处理

提高网站性能

AWS 博客：提高全球应用程序性能

使用 HTTPS 建立安全连接

AWS 博客： Amazon CloudFront：HTTPS 访问，增加一个节点，同时降价

用于配置私有内容的工具

除使用 CloudFront 提供的方法外，以下第三方工具均提供了用于为私有内容配置分配的 Web 表单。其中一些工具还提供了用于创建签名 URL 的 Web 表单。

- CloudBuddy：支持为私有内容配置分配并支持创建签名 URL。

  有关对 CloudFront 私有内容使用 CloudBuddy 的更多信息，请访问配置 CloudFront 分配和私有内容。

  该工具基于 CSS CorpLabs 在以下方面的研究成果：CloudFront 私有 URL 的 .NET 实现。
- Bucket Explorer：支持为私有内容配置分配

  有关对 CloudFront 私有内容使用 Bucket Explorer 的信息，请访问如何在存储桶中创建私有分配。
- CloudBerry：支持为私有内容配置分配并支持创建签名 URL。

  有关对 CloudFront 私有内容使用 CloudBerry 的信息，请访问如何使用 CloudBerry 为 CloudFront 流式处理配置私有内容。

有关私有内容的更多信息，请参阅 AWS 博客：Amazon CloudFront 新增功能：私有内容。

使用自定义源

AWS 博客：Amazon CloudFront 新增功能：自定义源

Amazon CloudFront 第三方工具概述

AWS 博客：CloudFront 管理工具综述

将 CloudFront 与内容管理系统结合使用

Drupal

- Drupal.org：CloudFront 安装
- DrupalModules.com：CloudFront Drupal 模块

Sitecore

- NTT 数据咨询服务：AWS CloudFront Sitecore 集成

WordPress

- om4.com：将 Amazon CloudFront 与 WordPress 和 WordPress MU 结合使用
- WordPress.org：W3 总缓存

- WordPress.org：简单的 Amazon S3 上传表单
- WordPress.org：OSSDL CDN Off-linker
- WordPress.org：我的 CDN
- Inquisiter.com：开通了 WordPress 博客的 Amazon CloudFront CDN

文档历史记录

下表描述了自上次发行 CloudFront 以来对文档所做的重要更改。

- API 版本：2013-11-11
- 文档最新更新时间：2013 年 12 月 18 日

更改	描述	修改日期
新功能	此版本的 CloudFront 引入了地理限制。如果您需要阻止选定国家/地区的用户访问您的内容，可以配置 CloudFront Web 分配来执行以下操作之一： • 仅当用户位于指定国家/地区的白名单中时才允许他们访问内容。 • 阻止位于指定国家/地区的黑名单中的用户访问内容。 有关更多信息，请参阅 限制您的内容的地理分配 (p. 42)。	2013 年 12 月 18 日
新功能	此版本的 CloudFront 引入了以下功能： • DELETE、OPTIONS、PATCH、POST 和 PUT 支持：现在，您可以在发送给 CloudFront 的请求中使用 `DELETE`、`OPTIONS`、`PATCH`、`POST` 和 `PUT` HTTP 方法。有关更多信息，请参阅 允许的 HTTP 方法 (p. 35)。 • 分配类型已重命名：CloudFront 下载分配现在称为 Web 分配，流分配现在称为 RTMP 分配。 • Web 分配的访问日志中的新列：CloudFront Web 分配的访问日志现在对每个请求新增了三列：`x-host-header`、`cs-protocol` 和 `cs-bytes`。有关更多信息，请参阅 Web 分配日志文件的格式 (p. 149)。	2013 年 10 月 15 日

更改	描述	修改日期
新功能	此版本的 CloudFront 引入了以下功能： • 自定义错误页面：现在，您可以提供包含您自己的品牌和内容的错误页面，而不是默认 HTTP 错误消息，如“404, page not found (404，页面未找到)”。当 Web 服务器不可用时，您还可以使用自定义错误页面来提供静态页面。有关更多信息，请参阅 自定义错误响应 (p. 70)。 • 可配置的错误响应缓存持续时间：此功能也称为错误缓存最短 TTL，它允许您指定您希望 CloudFront 将每个错误在 CloudFront 节点缓存多长时间。CloudFront 之前将所有错误响应缓存 5 分钟；现在您可以指定任何持续时间，从而控制在出现错误后 CloudFront 多久与源确认一次。有关更多信息，请参阅 自定义错误响应 (p. 70)。	2013 年 9 月 23 日
新功能	现在您可以在 CloudFront 备用域名（别名记录）中包含 * 通配符，例如，*.example.com。当您想要将针对某个域及其子域中的对象的所有要求发送到 CloudFront 分配时，这会非常有用。有关更多信息，请参阅 使用备用域名（别名记录） (p. 52)。	2013 年 9 月 18 日
更新的文档	添加了有关使用 Wowza Media Server 3.6 进行实时媒体流传输的文档。有关更多信息，请参阅 使用 Wowza Media Server 3.6 的实时 HTTP 流 (p. 199)。	2013 年 9 月 10 日
更新的文档	有关使用 Adobe Flash Media Server 进行实时媒体流传输的文档被替换为使用 Adobe Media Server 5.0 版进行实时媒体流传输的文档。有关更多信息，请参阅 使用 CloudFront 和 Adobe Media Server 5.0 的实时 HTTP 流 (p. 165)。	2013 年 7 月 31 日
新功能	此版本的 CloudFront 引入了以下功能： • 使用 AWS 签名版本 4 进行身份验证：如果您使用 CloudFront API 版本 2013-05-12 或更高版本，则必须使用 AWS 签名版本 4 对请求进行验证。有关更多信息，请参阅对 REST 请求进行身份验证 (p. 164)。 • CloudFront 备用域名的 SSL：CloudFront 现在支持使用 HTTPS 并支持在对象的 URL 中使用您自己的域名（例如 `http://www.example.com/image.jpg`）。有关更多信息，请参阅 使用备用域名和 HTTPS (p. 136)。 此外，Amazon Route 53 的同步版本引入了以下 CloudFront 相关功能： • CloudFront 分配的 Route 53 别名：Amazon Route 53 现在支持创建别名资源记录集，以便将 DNS 查询发送到 CloudFront 分配的备用域名。您可以对主域顶点的备用域名 (example.com) 和子域的备用域名 (www.example.com) 使用此功能。有关更多信息，请参阅 *Amazon Route 53 开发人员指南* 中的将查询发送到 Amazon CloudFront 分配。	2013 年 6 月 11 日

更改	描述	修改日期
新功能	此版本的 CloudFront 引入了以下功能： • AWS Management Console 中的私有内容字段：之前只能使用 CloudFront API 进行配置和更改的私有内容设置现在可以在AWS Management Console中进行配置或更改。这包括源访问身份和可信签署人的设置。此外，对有关私有内容的文档进行了重组和阐明。 有关更多信息，请参阅 通过 CloudFront 提供私有内容。 (p. 91)。 • AWS Management Console的改进：AWS Management Console 中的向导和对话框已经调整了大小，以简化在平板电脑上的观看，而不影响其他查看者看到的外观。此外，创建分配向导中的页数也已减少，以简化创建新分配的过程。	2012 年 9 月 27 日

新功能	此版本的 CloudFront 引入了以下功能：
	• Web 分配访问日志的改进：对于 Web 分配，CloudFront 访问日志现在包括以下字段：
	  • 每个查看者请求中的 Cookie 标头，包括名称值对和相关的属性。此字段为可选项。
	  • 请求的结果类型（例如，Hit、RefreshHit 或 Miss）。
	  • 唯一标识每个请求的标识符（CloudFront 请求 ID）。
	有关更多信息，请参阅 Web 分配日志文件的格式 (p. 149)。
	• Web 分配的 Cookie 支持：您现在可以选择是否希望 CloudFront 转发 Cookie 以及与源有关的 Cookie 属性。如果是，您还可以选择是转发所有 Cookie 还是只转发选定的 Cookie 列表。有关更多信息，请参阅 CloudFront 如何转发、缓存及记录 Cookie (p. 61)。
	• Web 和 RTMP 分配的价格级别：您现在可选择与您想为 CloudFront 服务支付的最高价对应的价格级别。如果您愿意接受让某些地理区域的查看者承受较高的延迟，以换取低成本，可以选择不包括所有 CloudFront 区域的价格级别。有关更多信息，请参阅 选择 CloudFront 分配的价格级别 (p. 54)。
		2012 年 9 月 5 日

新功能	此版本的 CloudFront 引入了以下功能： • 现在，您可使用 CloudFront 控制台使对象失效。有关更多信息，请参阅 使对象失效（仅 Web 分配） (p. 66)。 • CloudFront 控制台得到了更新，以更好地支持平板设备上的观看体验。	2012 年 6 月 22 日

更改	描述	修改日期
新功能	此版本的 CloudFront 为 Web 分配引入了以下功能： • 您可将查询字符串转发到您的源。有关更多信息，请参阅 CloudFront 如何转发、缓存及记录查询字符串参数 (p. 59)。 • 您最多可指定 10 个源。有关更多信息，请参阅 您创建或更新 Web 分配时指定的值 (p. 29)。 • 您可指定路径模式。有关更多信息，请参阅 您创建或更新 Web 分配时指定的值 (p. 29)。 此外，CloudFront 控制台也已经更新。有关更多信息，请参阅 创建 Web 分配 (p. 19) 和 创建 RTMP 分配 (p. 22)。 *Amazon CloudFront 入门指南* 已并入 *Amazon CloudFront 开发人员指南*，*Amazon CloudFront 开发人员指南* 进行了重组以提高可用性。	2012 年 5 月 13 日
更新的文档	对介绍如何处理对象的文档进行了重组和阐明。对于修订后的文档，请参阅使用对象 (p. 57)。	2012 年 4 月 4 日
新文档	添加了有关使用 Microsoft IIS Media Services 4.1 版进行实时流媒体传输的文档。有关更多信息，请参阅 使用 Amazon CloudFront 和 IIS Media Services 4.1 的实时平滑流 (p. 183)。	2012 年 4 月 1 日
更新的文档	有关使用 Adobe Flash Media Server 进行实时流媒体传输的文档已使用有关 Adobe Flash Media Server 4.5 版的信息进行了更新。 自 2013 年 7 月 31 日起，CloudFront 支持使用 Adobe Media Server 5.0 进行实时流媒体传输。有关更多信息，请参阅 使用 CloudFront 和 Adobe Media Server 5.0 的实时 HTTP 流 (p. 165)。	2012 年 3 月 29 日
新功能	此版本的 CloudFront 减少了 Web 分配的最小 TTL 值。如果您创建分配时未指定最小 TTL，CloudFront 会将最小 TTL 设置为 0 秒。有关更多信息，请转到以下文档： • CloudFront 产品页面 • Amazon S3 源的请求和响应行为 (p. 80)中的“缓存持续时间和最小 TTL” • 自定义源的请求和响应行为 (p. 83)中的“缓存持续时间和最小 TTL” • DistributionConfig 复杂类型中的 `CachingBehavior` 元素。	2012 年 3 月 15 日
更新的文档	关于使用 Adobe Flash Media Server 进行实时流媒体传输以及 geoblocking 的主题已从单独的文档移到本指南的CloudFront 教程 (p. 165)章节。	2012 年 2 月 2 日
新功能	此版本的 CloudFront 支持通过AWS Management Console创建具有自定义源的分配，限制只能通过 HTTPS 访问分配，并指定默认的根对象。有关更多信息，请转到 Amazon CloudFront 产品页面或参阅 *Amazon CloudFront 开发人员指南* 中的以下任一主题： • 创建 Web 分配 (p. 19) • 使用 HTTPS 连接访问您的对象 (p. 134) • 指定默认根对象（仅 Web 分配） (p. 74)	2011 年 4 月 27 日

更改	描述	修改日期
新功能	此版本的 CloudFront 包含与 AWS Identity and Access Management (IAM) 的集成。有关更多信息，请转到 *Amazon CloudFront 开发人员指南* 中的 Amazon CloudFront 产品页面或使用 IAM 控制对 CloudFront 资源的访问 (p. 141)。	2011 年 3 月 10 日
新功能	此版本的 CloudFront 包括支持自定义源的新 API。有关更多信息，请转到 *Amazon CloudFront 开发人员指南* 中的 Amazon CloudFront 产品页面或创建 Web 分配 (p. 19)。	2010 年 9 月 9 日
新功能	此版本的 CloudFront 包括使对象失效的新 API。有关更多信息，请转到 *Amazon CloudFront 开发人员指南* 中的 Amazon CloudFront 产品页面或使对象失效（仅 Web 分配）(p. 66)。	2010 年 8 月 31 日
新功能	CloudFront 现在支持为您的分配指派默认根对象的能力。有关更多信息，请参阅 指定默认根对象（仅 Web 分配）(p. 74)。	2010 年 8 月 5 日
新功能	HTTP 分配的访问记录现在包括查询字符串参数的字段。有关更多信息，请参阅 Web 分配日志文件的格式 (p. 149)。	2010 年 7 月 14 日
新功能	添加了对使用 HTTPS 的安全连接的支持。有关更多信息，请参阅 使用 HTTPS 连接访问您的对象 (p. 134)。	2010 年 6 月 7 日
新功能	添加了 RTMP 内容的日志记录。有关更多信息，请参阅 RTMP 分配日志文件的格式 (p. 151)。	2010 年 5 月 13 日
新功能	将对象可在边缘服务器上保留的最短时间从 24 小时减少到 1 小时。但默认情况下保留 24 小时。有关更多信息，请参阅过期 (p. 63)。	2010 年 4 月 13 日
新功能	添加了新功能，以通过实时消息传递协议 (RTMP) 提供私有流媒体内容，并防止内容被下载。有关更多信息，请参阅 通过 CloudFront 提供私有内容。 (p. 91)。	2010 年 3 月 28 日
新功能	添加了新功能，以通过实时消息传递协议 (RTMP) 连接传送流媒体内容。有关更多信息，请参阅 创建 RTMP 分配 (p. 22)。	2009 年 12 月 15 日
新功能	添加了新功能，以限制访问通过 HTTP 传送的内容。有关更多信息，请参阅 通过 CloudFront 提供私有内容。 (p. 91)。	2009 年 11 月 11 日
新指南	我们已经将 API 参考资料分散到其各自的指南中。*Amazon CloudFront 开发人员指南* 包含有关如何使用 CloudFront 的一般信息，Auto Scaling API 参考 包含有关控制 API 请求、响应和错误的详细信息。	2009 年 11 月 11 日