

AWS对于各项功能均有详细的官方文档可供参考，但官方文档过于繁杂，本文将就CloudFront服务的使用流程作简单说明。

目的：

为站点www.dancen.com开通CloudFront CDN服务。

步骤：

1. 打开CloudFront服务之后，首先需要选择分发类型。分发类型分为一般的静态文件分发和流媒体分发，即Web和RTMP，我的站点即选择Web类型。

Step 1: Select delivery method

Step 2: Create distribution

Select a delivery method for your content. ?

Web

Create a web distribution if you want to:

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

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

Get Started

RTMP

Create an RTMP distribution to speed up distribution of your streaming media files using Adobe Flash Media Server's RTMP protocol. An RTMP distribution allows an end user to begin playing a media file before the file has finished downloading from a CloudFront edge location. Note the following:

- To create an RTMP distribution, you must store the media files in an Amazon S3 bucket.
- To use CloudFront live streaming, create a web distribution.

Get Started

Cancel

2. CDN回源设定。

a. 首先要填写源站地址Origin Domain Name，源站可以是一个AWS S3，也可是是一个普通web站点。对于web站点，你要填写的便是该web站点的域名，不支持直接填写IP；需要注意的是，该域名不能与站点提供给用户访问的域名一致，而是一个单独的回源域名。例如对于我的站点，其域名为www.dancen.com，我新注册了一个域名wwwcdn.dancen.com用于CDN回源。为什么回源域名不能与站点域名一致呢，很好理解，以我的站点为例，当用户访问站点www.dancen.com时，通过DNS系统查询得到站点的CNAME记录，CDN再通过对CNAME的解析到达边缘节点；当需要回源时，CDN又会访问回源域名，如果回源域名也是www.dancen.com，那就形成了一个循环，www.dancen.com -> CDN -> www.dancen.com，而且这是个死循环。

b. 其次，需要选择回源协议，对安全要求不高的话，选择http only即可。

Step 1: Select delivery method

Step 2: Create distribution

Create Distribution



Origin Settings

Origin Domain Name

Origin Path

Origin ID

Origin SSL Protocols ☒ TLSv1.2
☒ TLSv1.1
☒ TLSv1
☐ SSLv3

Origin Protocol Policy ☒ HTTP Only
☐ HTTPS Only
☐ Match Viewer

Origin Response Timeout

Origin Keep-alive Timeout

HTTP Port

HTTPS Port

Origin Custom Headers Header Name

3. 缓存设定。

a. 需要指定用户访问的协议，http和https、重定向http到https、仅https，根据需要选择。

b. http方法，默认选择GET，HEAD即可。

c. 需要注意的是对象缓存Object Caching的设定，该设定用来指定 CloudFront 为 Web 分配缓存对象的时长。

Use Origin Cache Headers选项表示以源站http header中Cache-Control max-age 或 Cache-Control s-maxage 指令或者 Expires 标头字段的设定为准；

Customize选项中，最短、最长和默认 TTL 值来控制 CloudFront 在缓存中保留对象的时长 (以秒为单位)，超过该时长后才会将另一个请求转发到源。标头值还确定浏览器在缓存中保留对象的时长，超过该时长后才会将另一个请求转发到 CloudFront。

d. Query String Forwarding and Caching设定。CDN中缓存的对象是以查询的URL来区分的，该项的默认选择为NONE，表示忽略URL 中? 之后的参数，有效提高文件缓存命中率，提升分发效率，这种情况下，

`http://www.dancen.com/t.png?v1`

`http://www.dancen.com/t.png?v2`

会被认为是同样的请求，当t.png已经被CDN缓存，并且没有过期时，即使服务器上的t.png被修改了，用户也获取不到新版的t.png。

虽然AWS官方推荐使用不同的文件名来对文件进行版本控制，但为了管理方便，我打算通过参数来对缓存进行版本控制，因此该项我设置为保留所有参数，即Forward all, cache based on all，这种情况下，

`http://www.dancen.com/t.png?v1`

`http://www.dancen.com/t.png?v2`

是两个不同的请求，当我修改了服务器上的t.png时，客户端只需要修改URL的参数，就能够获取到最新的t.png。

Step 1: Select delivery method

Step 2: Create distribution

Default Cache Behavior Settings

Path Pattern

Default (*)

Viewer Protocol Policy

☒ HTTP and HTTPS

☐ Redirect HTTP to HTTPS

☐ HTTPS Only

Allowed HTTP Methods

☒ GET, HEAD

☐ GET, HEAD, OPTIONS

☐ GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE

Field-level Encryption Config

Cached HTTP Methods

GET, HEAD (Cached by default)

Cache Based on Selected Request Headers

None (Improves Caching)

Learn More

Object Caching

☒ Use Origin Cache Headers

☐ Customize

Learn More

Minimum TTL

0

Maximum TTL

31536000

Default TTL

86400

Forward Cookies

None (Improves Caching)

Query String Forwarding and Caching

Forward all, cache based on all

Smooth Streaming

☐ Yes

☒ No

Restrict Viewer Access (Use Signed URLs or Signed Cookies)

☐ Yes

☒ No

Compress Objects Automatically

☐ Yes

☒ No

Learn More

Lambda Function Associations

Event Type

Lambda Function ARN

4. 分发设定。
- a. 价格类别，价格类别由分发区域决定，最高档即全球分发，最低档即只在AWS的大本营北美进行分发。

b. 源站域名Alternate Domain Names，即我的站点提供给用户访问的域名，www.dancen.com，而该域名的CNAME，便要设定为最后CloudFront为我们生成的CDN域名：xxxxxxx.cloudfront.net。

c. SSL证书设置。
- 默认的ColudFront证书Default CloudFront Certificat。如果你让用户通过最后CloudFront为我们生成的CDN域名：xxxxxxx.cloudfront.net直接访问你的站点，那选择该选项。

自定义SSL证书。如果你让用户通过你自己的域名，例如www.dancen.com访问你的站点，那你就得为CloudFront提供该站点的证书，该证书可以通过ACM（AWS Certificate Manager）导入，具体流程参照本文最厚的的附文部分。

d. 创建分发，点击Create Distribution按钮即可。

Step 1: Select delivery method

Step 2: Create distribution

Distribution Settings

Price Class

Use All Edge Locations (Best Perf)

AWS WAF Web ACL

None

Alternate Domain Names (CNAMEs)

www.dancen.com

SSL Certificate

Default CloudFront Certificate (*.cloudfront.net)

Choose this option if you want your users to use HTTPS or HTTP to access your content with t CloudFront domain name (such as https://d1111111abcdef8.cloudfront.net/logo.jpg). Important: If you choose this option, CloudFront requires that browsers or devices support TLS later to access your content.

Custom SSL Certificate (example.com):

Choose this option if you want your users to access your content by using an alternate domain name, such as https://www.example.com/logo.jpg. You can use a certificate stored in AWS Certificate Manager (ACM) in the US East (N. Virginia) Region, or you can use a certificate stored in IAM.

7eab8538-f216-4bd

Request or Import a Certificate with ACM

Learn more about using custom SSL/TLS certificates with CloudFront.

Learn more about using ACM.

Custom SSL Client Support

Only Clients that Support Server Name Indication (SNI)

CloudFront serves your content over HTTPS only to clients that support SNI. Older browsers ar other clients that do not support SNI can not access your content over HTTPS.

Learn More

All Clients (\$600/month prorated charge applies. Learn about pricing.)

CloudFront allocates dedicated IP addresses at each CloudFront edge location to serve your co over HTTPS. Any client can access your content.

Learn More

Security Policy

TLSv1

TLSv1_2016

TLSv1.1_2016 (recommended)

TLSv1.2_2018

See the list of protocols and ciphers that CloudFront uses for each security policy.

Supported HTTP Versions

HTTP/2, HTTP/1.1, HTTP/1.0

HTTP/1.1, HTTP/1.0

Default Root Object

Logging

On

Off

Bucket for Logs

Log Prefix

Cookie Logging

On

Off

Enable IPv6

☒

Cancel

Back

Create Distribution

5 获取CloudFront为我们生成的CDN域名。

创建分发后，待Status变为Deployed，说明CloudFront配置生效，这个过程可能耗费20分钟左右。

CloudFront会生成一个域名Domain Name: xxxxxxx.cloudfront.net，我们需要在域名提供商的DNS系统把源站域名，如www.dancen.com的CNAME记录设定为该域名。在CMD窗口执行nslookup命令检查你的域名，如www.dancen.com，能否解析出CDN提供的IP，确认CNAME设定是否生效，生效后CDN加速便算是开启成功，使用你的域名，如www.dancen.com测试通过即可。

CloudFront Distributions

Create Distribution

Distribution Settings

Delete

Enable

Disable

Viewing:

Any Delivery Method

Any State

	Delivery Method	ID	Domain Name	Origin	CNAMEs	Status
<input type="checkbox"/>	Web	E1P9PHJTHC3BIT	1017123456789.cloudfront.net	wwwcdn.guangchaiyuan.c	www.guangchaiyuan.	Deployed

```
C:\Users\>nslookup 1017123456789.cloudfront.net
服务器: UnKnown
Address: 10.17.1.2

非权威应答:
名称:    all.lv2.qnydns.net
Addresses: 14.119.81.23
          113.105.169.16
          113.105.169.17
          113.113.95.21
          113.113.95.22
          14.119.81.22
Aliases: 1017123456789.cloudfront.net
         1017123456789.cloudfront.net
         s113.105.169.16.qiniudns.com
         s113.105.169.17.qiniudns.com
         s113.113.95.21.qiniudns.com
         s113.113.95.22.qiniudns.com
```

附：

通过AWS的ACM（AWS Certificate Manager）服务导入站点SSL证书。ACM服务可以直接请求证书，也可以导入证书，这里假设你的源站已经使用https，有了SSL证书，因此只讲述导入证书的步骤。

1. 打开ACM服务，点击导入证书。

请求证书

导入证书

操作

正在查看 1 证书的 1 至 1

	名称	域名	其它域名	状态	类型	使用中?
<input type="checkbox"/>		www.guangchaiyuan.cn	www.guangchaiyuan.cn	已颁发	已导入	是

正在查看 1 证书的 1 至 1

到你的站点服务器，查看证书内容，并填入ACM。例如fullchain.pem文件即为证书，打开之，第一个begin-end即为证书正文，第二个begin-end即为证书链；privkey.pem文件即为私匙。填写完毕后，点击审核并导入即可。

步骤 1：导入证书

选择证书

```
PRZ+G6267njper0Y+WKY+HPWL.7w6lusaC6GSSW3pBM+X+QTUAKGRO6DNG
KOqkqm5TTH0HG6DJK5nh6/DNFu0Dg==
-----END CERTIFICATE-----
```

取消 审核并导入

<https://blog.csdn.net/Dancen/article/details/79524960>

<https://blog.csdn.net/Dancen/article/details/79524960>