**Domain: Networking and Content Delivery**

**Topic: CloudFront**

**Task: Create simple CloudFront distribution**

**Service description**

Amazon CloudFront is a web service that speeds up distribution of your static and dynamic web content, such as .html, .css, .js, and image files, to your users. CloudFront delivers your content through a worldwide network of data centers called edge locations. When a user requests content that you're serving with CloudFront, the request is routed to the edge location that provides the lowest latency (time delay), so that content is delivered with the best possible performance.

If the content is already in the edge location with the lowest latency, CloudFront delivers it immediately.

If the content is not in that edge location, CloudFront retrieves it from an origin that you've defined—such as an Amazon S3 bucket, a MediaPackage channel, or an HTTP server (for example, a web server) that you have identified as the source for the definitive version of your content.

As an example, suppose that you're serving an image from a traditional web server, not from CloudFront. For example, you might serve an image, sunsetphoto.png, using the URL http://example.com/sunsetphoto.png.

Your users can easily navigate to this URL and see the image. But they probably don't know that their request is routed from one network to another—through the complex collection of interconnected networks that comprise the internet—until the image is found.

CloudFront speeds up the distribution of your content by routing each user request through the AWS backbone network to the edge location that can best serve your content. Typically, this is a CloudFront edge server that provides the fastest delivery to the viewer. Using the AWS network dramatically reduces the number of networks that your users' requests must pass through, which improves performance. Users get lower latency—the time it takes to load the first byte of the file—and higher data transfer rates.

You also get increased reliability and availability because copies of your files (also known as objects) are now held (or cached) in multiple edge locations around the world.

You create a CloudFront distribution to tell CloudFront where you want content to be delivered from, and the details about how to track and manage content delivery. Then CloudFront uses computers—edge servers—that are close to your viewers to deliver that content quickly when someone wants to see it or use it.

 
    How CloudFront works 
   

1. You specify origin servers, like an Amazon S3 bucket or your own HTTP server, from which CloudFront gets your files which will then be distributed from CloudFront edge locations all over the world. An origin server stores the original, definitive version of your objects. If you're serving content over HTTP, your origin server is either an Amazon S3 bucket or an HTTP server, such as a web server. Your HTTP server can run on an Amazon Elastic Compute Cloud (Amazon EC2) instance or on a server that you manage; these servers are also known as custom origins.
2. You upload your files to your origin servers. Your files, also known as objects, typically include web pages, images, and media files, but can be anything that can be served over HTTP. If you're using an Amazon S3 bucket as an origin server, you can make the objects in your bucket publicly readable, so that anyone who knows the CloudFront URLs for your objects can access them. You also have the option of keeping objects private and controlling who accesses them. See [Serving private content with signed URLs and signed cookies](https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/PrivateContent.html).
3. You create a CloudFront distribution, which tells CloudFront which origin servers to get your files from when users request the files through your web site or application. At the same time, you specify details such as whether you want CloudFront to log all requests and whether you want the distribution to be enabled as soon as it's created.
4. CloudFront assigns a domain name to your new distribution that you can see in the CloudFront console, or that is returned in the response to a programmatic request, for example, an API request. If you like, you can add an alternate domain name to use instead.
5. CloudFront sends your distribution's configuration (but not your content) to all of its edge locations or points of presence (POPs)— collections of servers in geographically-dispersed data centers where CloudFront caches copies of your files.

**Usecases**

* *Accelerate static website content delivery*CloudFront can speed up the delivery of your static content (for example, images, style sheets, JavaScript, and so on) to viewers across the globe. By using CloudFront, you can take advantage of the AWS backbone network and CloudFront edge servers to give your viewers a fast, safe, and reliable experience when they visit your website.
* *Serve video on demand or live streaming video*For video on demand (VOD) streaming, you can use CloudFront to stream in common formats such as MPEG DASH, Apple HLS, Microsoft Smooth Streaming, and CMAF, to any device. For broadcasting a live stream, you can cache media fragments at the edge, so that multiple requests for the manifest file that delivers the fragments in the right order can be combined, to reduce the load on your origin server.
* *Encrypt specific fields throughout system processing*When you configure HTTPS with CloudFront, you already have secure end-to-end connections to origin servers. When you add field-level encryption, you can protect specific data throughout system processing in addition to HTTPS security, so that only certain applications at your origin can see the data.
* *Customize at the edge*Running serverless code at the edge opens up a number of possibilities for customizing the content and experience for viewers, at reduced latency. For example, you can return a custom error message when your origin server is down for maintenance, so viewers don't get a generic HTTP error message. Or you can use a function to help authorize users and control access to your content, before CloudFront forwards a request to your origin.

**Problem to Be Solved**

*Imagine you have a static web site deployed at London and you have users all around the world. And your users when they go to your website, they’re going directly to London and pull that content down. Basically, if you have your web servers in London and you've got all your media files, so your images, your videos, etc., individual has to go and, essentially, they're connecting into that server across the internet, and they're pulling those files down, literally across oceans, and it can take time to load.*

## Explanation of the Solution

[*CloudFront introduction*](https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Introduction.html)

[*Demo of simple CloudFront distribution*](https://www.youtube.com/watch?v=KIltfPRpTi4&ab_channel=AmazonWebServices)

## Implementation Details

* *Create simple s3 bucket with default settings*
* *Upload any picture to this bucket*
* *Make this picture publicly accessible*
* *Create a distribution*
  + *Choose bucket you created as Origin Domain Name* Graphical user interface, text, application, chat or text message

    Description automatically generated
  + *Restrict bucket access* Graphical user interface, text, application, chat or text message

    Description automatically generated
  + *Redirect all http traffic to https* Text

    Description automatically generated
  + *Leave all settings default*
* *Visit distribution you created and ensure that file is available*

## Benefits / Outcomes / Pros and Cons / Summary

* *Global Scaled Network for Fast Content Delivery*
* *Security at the Edge*
* *Highly Programmable and Secure Edge Computing*
* *Deep Integration with AWS*
* *Cost-Effective*

## Pricing

[*CloudFront pricing explanation*](https://aws.amazon.com/cloudfront/pricing/)

## Tearing down

* *Disable cloudfront distribution*
* *Remove cloudfront distribution*
* *Remove s3 bucket*

## Limits

| **Entity** | **Default quota** |
| --- | --- |
| Data transfer rate per distribution | 150 Gbps (soft) |
| Requests per second per distribution | 250,000 (soft) |
| Tags that can be added to a distribution | 50 |
| Files that you can serve per distribution | No quota |
| Maximum length of a request, including headers and query strings, but not including the body content | 20,480 bytes |
| Maximum length of a URL | 8,192 bytes |

[*More info about cloudfront quotas*](https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/cloudfront-limits.html)