Cloudfront

CloudFront is a CDN service provided by AWS.

Components:

Origin Servers: Origin Servers are the actual source of the objects that needs to be distributed through Cloudfront. A origin server can be a S3 bucket or a custom HTTP Server.
Cloud Front works only with **HTTP** and **RTMP** protocols.
RTMP (Real Time Messaging Protocol) : Used in Adobe Flash.

If your origin server is S3, you need to make objects publicly accessible so that Cloudfront can access them. We will see a way around for this in "Serving Private Content through Cloudfront".

Configurations are maintained per distribution basis. So, for each distribution a separate domain name is allocated via cloudfront. The domain name is of the format:
http://d111111abcdef8.cloudfront.net/logo.jpg

However, we can setup our custom domain with a CNAME record, so that we can point our custom domain to the domain provided by cloudfront distribution. This way, we can point something like:
http://example.com/logo.jpg to http://d111111abcdef8.cloudfront.net/logo.jpg

Edge Locations: These are the front-facing servers, which receives the client requests for objects and decides whether the request needs to be served from their local cache or from origin servers.

In case, the request needs to be served from origin server, following happens:

1. Edge location sends the request for object to origin server.
2. As soon as **first byte of response** is received from origin server, the edge location begins to transfer the content to Client.
3. Also, it caches the content at edge location.

Regional Edge Caches

Regional Edge Caches is a layer between Edge locations and origin servers. They are like secondary caches. If an object is not found at Edge location, it is first searched at Regional Edge Caches before making a call to origin server.

These caches comes into role for objects which are not frequently accessed. As, due to less frequent access they are generally evicted from Edge locations before their expiry time, such objects are generally retained at Regional caches as they have much larger cache size than edge locations.
- Regional Caches are **enabled by default** for all cloudfront distributions.
- There is **no cost** of this feature.
- Cache Invalidation requests, Invalidates cache from both Edge locations and Regional Caches.

NOTE: Regional Edge caches are **not supported for S3** origins, They are only supported for Custom HTTP servers.

Lambda@Edge

This is a feature that allows you to execute custom lambda function for certain Cloudfront events.

There are **4 events** in the lifecycle of a object request as identified by cloudfront:

1. **CloudFront Viewer Request:**
   - The event is triggered when, cloudfront receives a request from viewer and before it checks whether the object can be
   served from local cache.
2. **CouldFront Origin Request:**
   - The event is triggered only when, cloudfront decides to forward the request to origin server.
   If the object is found in cache the event is not triggered.
3. **CloudFront Origin Response:**
   - The event is triggered when, cloudfront receives a response from origin and before it caches the content.
4. **CloudFront Viewer Response:**
   - The event is triggered before returning the object to viewer.
   The event is triggered regardless of the fact that object was found in cache or not.

Service Private Content

This method is used when we are to serve data only to authenticated users. An example could be that only users that have paid for the service can view the video/image.
When using cloudfront private content needs to be protected at two ends:

a. At cloudfront edge locations.
b. At origin servers.

At Cloudfront Edge locations:

Cloudfront provides the use of Signed URLS and Signed Cookies. Your application needs to provide **SignedUrls** or **Signed cookies** to the authenticated users. Such that the requests made by authenticated users contains signed URL or Signed Cookie, instead of the actual resource URL.
Cloudfront then compares Signed and Unsigned portions of request and if they match then only the request is served.

At Origin Server:

If the Origin Server is a S3 bucket, you need to create a special user for Cloudfront access. This special user is called "**Origin Access Identity**". Now, only this user is Granted access to S3 objects.

For custom servers, we need to device our own methods. One way could be to allow access, only if the request has a valid auth token.

Working with Objects

You can configure cloudfront to distinguish objects based on **query params**, **Headers** and **Cookies**.
A single distribution has **40Gbps** network bandwidth and can support no more than **1 Lakh** requests/second.

Versioning vs Invalidation

- The best way to update an object or to remove an object is by using versioning of objects.
- Invalidation of objects has limitations, takes time and is not purely free.
- Multiple Objects can be invalidated in single call by using wildcard(*) in object path.

Invalidation Limits:
- AWS can invalidate only **3000** objects at one time for a distribution. Either, you make single request of 3000 objects invalidations or 10 requests with 300 objects each it does not matter.
- Only **1000** invalidation paths are supported per month for free tier. A path can be of a single object or with a wildcard that represents thousands of objects.

CloudFront FAQS:
1. Cloudfront caches each object for **24 hours** in case no TTL is supplied in headers. The minimum expiration time is 0 sec.
2. The transfer from Edge location to client begins as soon as first byte is received from origin server or Regional Caches.
3. S3 origin does not support Regional Edge caches. They are supported only for custom origins like: HTTP server.
4. Only GET Requests are cached and served by edge locations. PUT/POST/PATCH/OPTIONS/DELETE etc are served directly via origin servers.
5. CloudFront supports **gzip compression** as well.
6. Cloudfront supports range requests. Range Requests are GET requests for objects which asks for only a part of object instead of whole object.