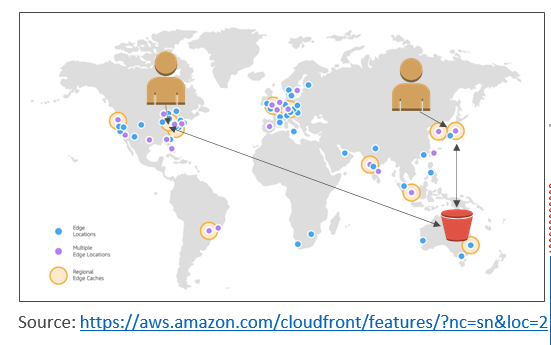
**AWS CloudFront**

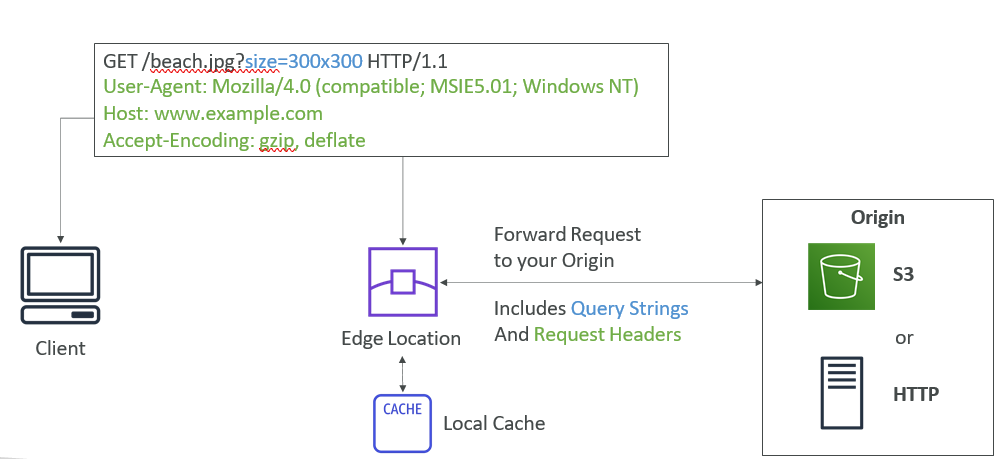
* Content Delivery Network (CDN)
* Improves read performance, content is cached at the edge
* 216 Point of Presence globally (edge locations)
* DDoS protection, integration with Shield, AWS Web Application Firewall
* Can expose external HTTPS and can talk to internal HTTPS backends



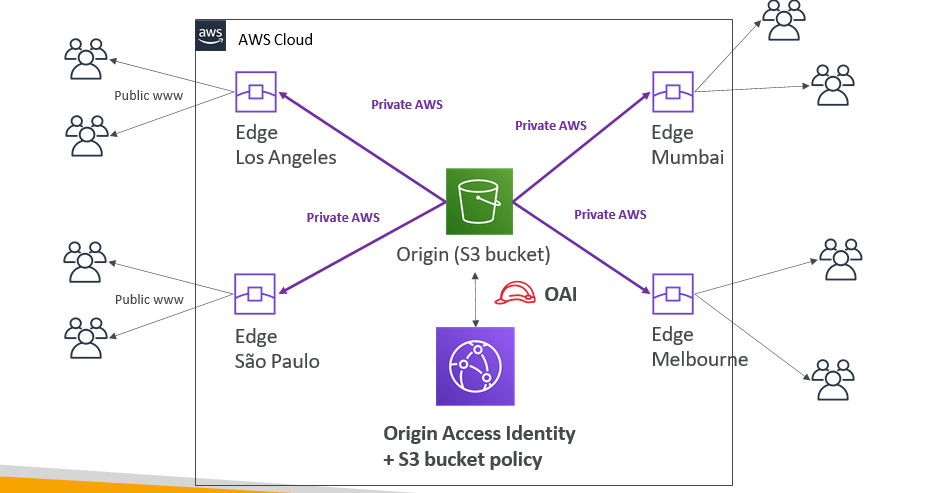
**CloudFront – Origins**

* S3 bucket
  + For distributing files and caching them at the edge
  + Enhanced security with CloudFront Origin Access Identity (OAI)
  + CloudFront can be used as an ingress (to upload files to S3)
* Custom Origin (HTTP)
  + Application Load Balancer
  + EC2 instance
  + S3 website (must first enable the bucket as a static S3 website)
  + Any HTTP backend you want

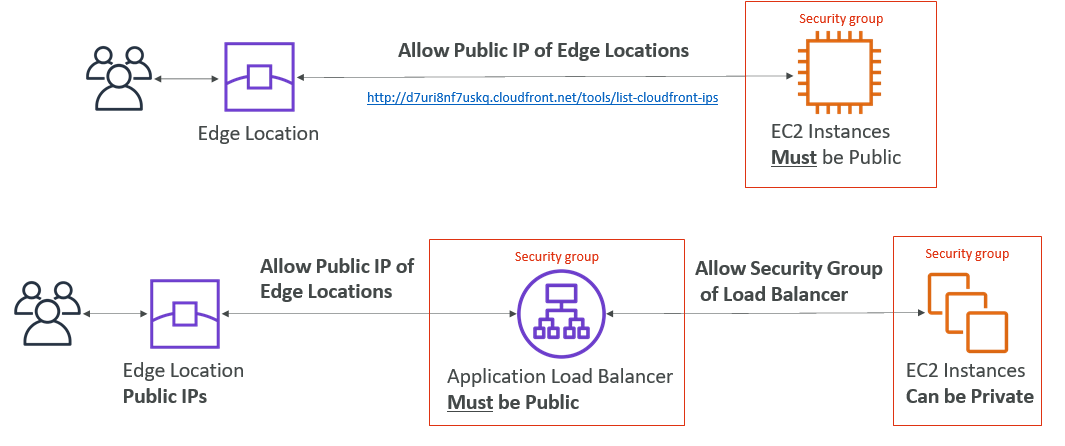
**CloudFront at a high level**



**CloudFront – S3 as an Origin**



**CloudFront – ALB or EC2 as an origin**



**CloudFront Geo Restriction**

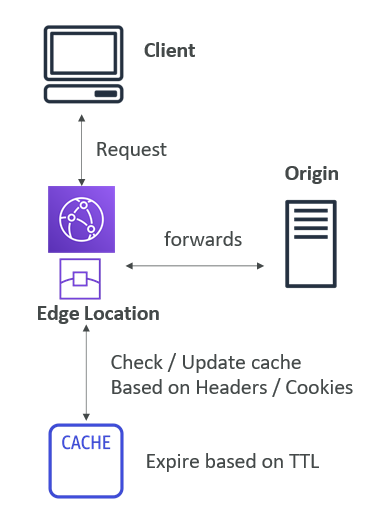
* You can restrict who can access your distribution
  + Whitelist: Allow your users to access your content only if they're in one of the countries on a list of approved countries.
  + Blacklist: Prevent your users from accessing your content if they're in one of the countries on a blacklist of banned countries.
* The “country” is determined using a 3rd party Geo-IP database
* Use case: Copyright Laws to control access to content

**CloudFront vs S3 Cross Region Replication**

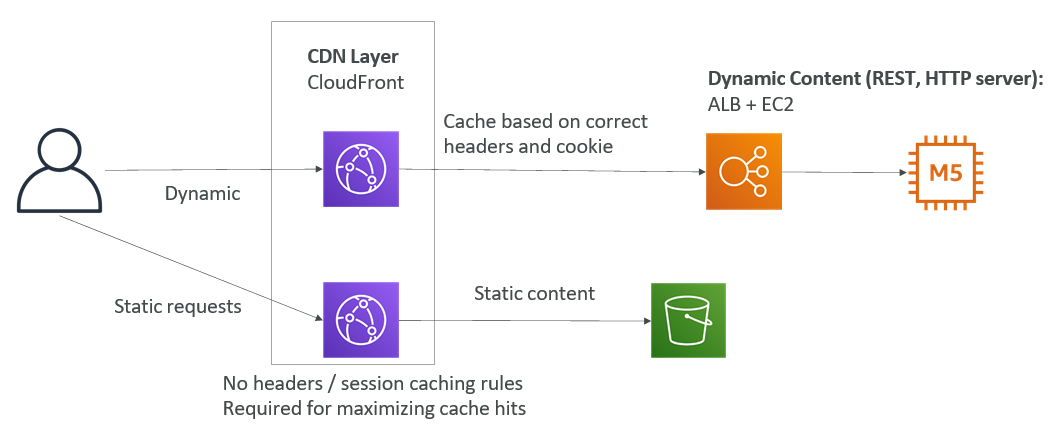
* CloudFront:
  + Global Edge network
  + Files are cached for a TTL (maybe a day)
  + Great for static content that must be available everywhere
* S3 Cross Region Replication:
  + Must be setup for each region you want replication to happen
  + Files are updated in near real-time
  + Read only
  + Great for dynamic content that needs to be available at low-latency in few regions

**CloudFront Caching**

* Cache based on
  + Headers
  + Session Cookies
  + Query String Parameters
* The cache lives at each CloudFront Edge Location
* You want to maximize the cache hit rate to minimize requests on the origin
* Control the TTL (0 seconds to 1 year), can be set by the origin using the Cache- Control header, Expires header…
* You can invalidate part of the cache using the CreateInvalidation API



**CloudFront – Maximize cache hits by separating static and dynamic distributions**



**CloudFront Geo Restriction**

* You can restrict who can access your distribution
  + Whitelist: Allow your users to access your content only if they're in one of the countries on a list of approved countries.
  + Blacklist: Prevent your users from accessing your content if they're in one of the countries on a blacklist of banned countries.
* The “country” is determined using a 3rd party Geo-IP database
* Use case: Copyright Laws to control access to content

**CloudFront and HTTPS**

* Viewer Protocol Policy:
  + Redirect HTTP to HTTPS
  + Or use HTTPS only
* Origin Protocol Policy (HTTP or S3):
  + HTTPS only
  + Or Match Viewer

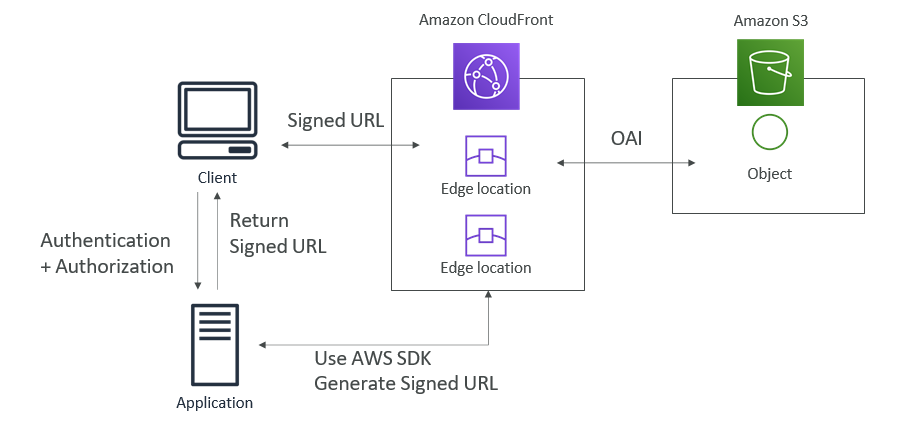
(HTTP => HTTP & HTTPS => HTTPS)

* Note:
  + S3 bucket “websites” don’t support HTTPS

**CloudFront Signed URL / Signed Cookies**

* You want to distribute paid shared content to premium users over the world
* To Restrict Viewer Access, we can create a CloudFront Signed URL / Cookie
* How long should the URL be valid for?
  + Shared content (movie, music): make it short (a few minutes)
  + Private content (private to the user): you can make it last for years
* Signed URL = access to individual files (one signed URL per file)
* Signed Cookies = access to multiple files (one signed cookie for many files)

**CloudFront Signed URL Diagram**



**CloudFront Signed URL vs S3 Pre-Signed URL**

* CloudFront Signed URL:
* Allow access to a path, no matter the origin
* Account wide key-pair, only the root can manage it
* Can filter by IP, path, date, expiration
* Can leverage caching features
* S3 Pre-Signed URL:
* Issue a request as the person who pre-signed the URL
* Uses the IAM key of the signing IAM principal
* Limited lifetime

**CloudFront Signed URL Process**

* Two types of signers:
* Either a trusted key group (recommended)
  + Can leverage APIs to create and rotate keys (and IAM for API security)
* An AWS Account that contains a CloudFront Key Pair
  + Need to manage keys using the root account and the AWS console
  + Not recommended because you shouldn’t use the root account for this
* In your CloudFront distribution, create one or more trusted key groups
* You generate your own public / private key
  + The private key is used by your applications (e.g. EC2) to sign URLs
  + The public key (uploaded) is used by CloudFront to verify URLs

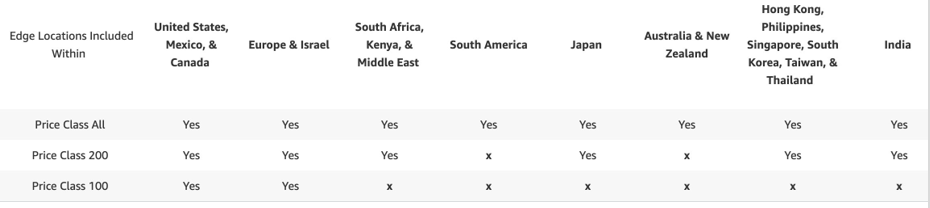
**CloudFront - Pricing**

* CloudFront Edge locations are all around the world
* The cost of data out per edge location varies

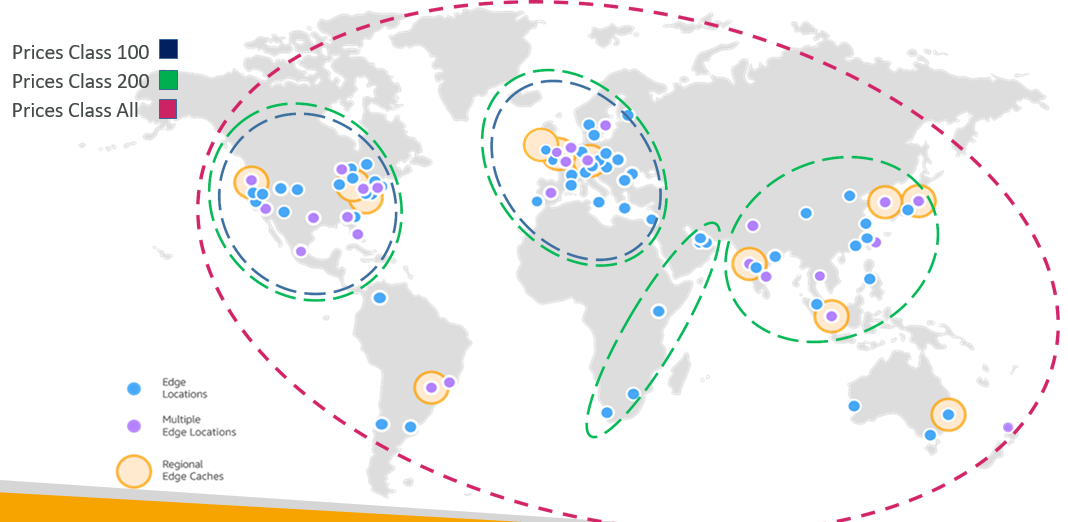


**CloudFront – Price Classes**

* You can reduce the number of edge locations for cost reduction
* Three price classes:
  + Price Class All: all regions – best performance
  + Price Class 200: most regions, but excludes the most expensive regions
  + Price Class 100: only the least expensive regions

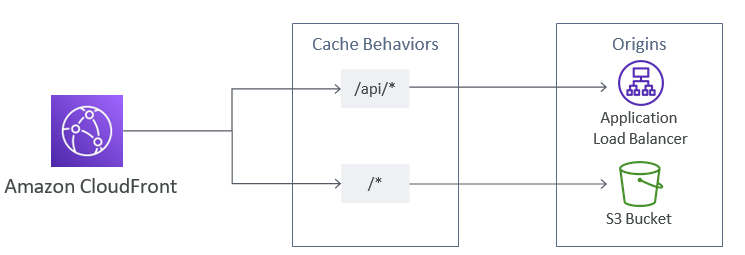


**CloudFront - Price Class**



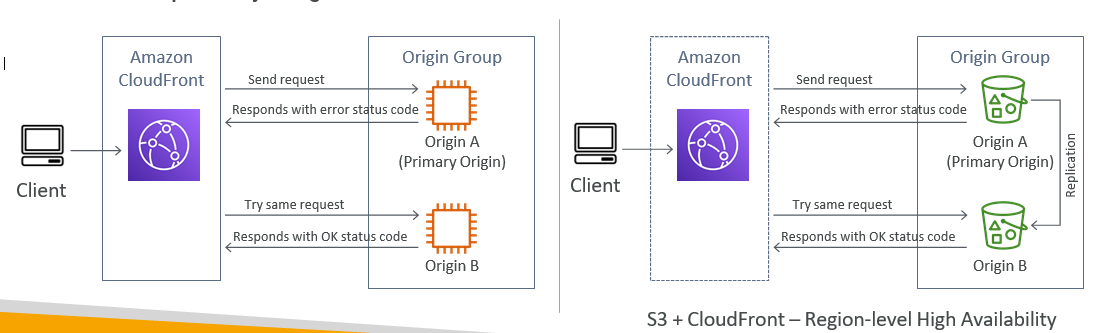
**CloudFront – Multiple Origin**

* To route to different kind of origins based on the content type
* Based on path pattern:
  + /images/\*
  + /api/\*
  + /\*



**CloudFront – Origin Groups**

* To increase high-availability and do failover
* Origin Group: one primary and one secondary origin
* If the primary origin fails, the second one is used



**CloudFront – Field Level Encryption**

* Protect user sensitive information through application stack
* Adds an additional layer of security along with HTTPS
* Sensitive information encrypted at the edge close to user
* Uses asymmetric encryption
* Usage:
  + Specify set of fields in POST requests that you want to be encrypted (up to 10 fields)
  + Specify the public key to encrypt them

