**Content Delivery Network**

* You want to deliver content to your global audience
* Content Delivery Networks distribute content to multiple edge locations around the world
* AWS provides 200+ edge locations around the world
* Provides high availability and performance

**Amazon CloudFront**

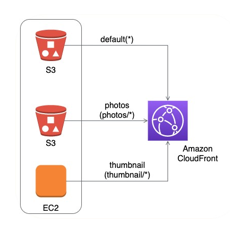
* How do you enable serving content directly from AWS edge locations?
  + Amazon CloudFront (one of the options)
* Serve users from nearest edge location (based on user location)
* Source content can be from S3, EC2, ELB and External Websites
* If content is not available at the edge location, it is retrieved from the origin server and cached
* No minimum usage commitment
* Provides features to protect your private content

**Amazon CloudFront**

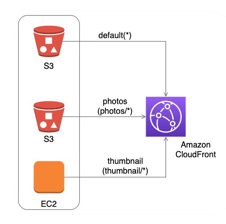
* Use Cases
  + Static web apps. Audio, video and software downloads. Dynamic web apps
  + Support media streaming with HTTP and RTMP
* Integrates with
  + AWS Shield to protect from DDoS attacks
  + AWS Web Application Firewall (WAF) to protect from SQL injection, cross- site scripting, etc
* Cost Benefits
  + Zero cost for data transfer between S3 and CloudFront
  + Reduce compute workload for your EC2 instances

**Amazon CloudFront Distribution**

* Create a CloudFront Distribution to distribute your content to edge locations
  + DNS domain name - example abc.cloudfront.com
  + Origins - Where do you get content from? S3, EC2, ELB, External Website
  + Cache-Control
    - By default objects expire after 24 hours
    - Customize min, max, default TTL in CloudFront distribution
    - (For file level customization) Use Cache-Control max-age and Expires headers in origin server
* You can configure CloudFront to only use HTTPS (or) use HTTPS for certain objects
  + Default is to support both HTTP and HTTPS
  + You can configure CloudFront to redirect HTTP to HTTPS

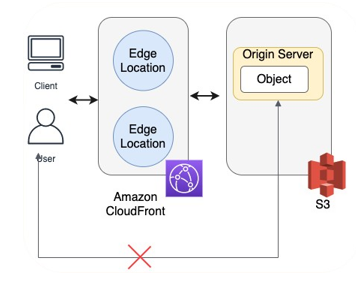


**Amazon CloudFront - Cache Behaviors**



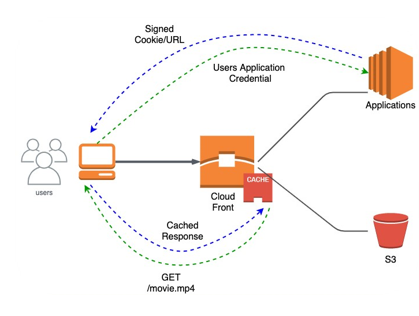
* Configure different CloudFront behavior for different URL path patterns from same origin
  + Path pattern(can use wild cards - \*.php, \*.jsp),
  + Do you want to forward query strings? Should we use https?
  + TTL

**Amazon CloudFront - Private content**



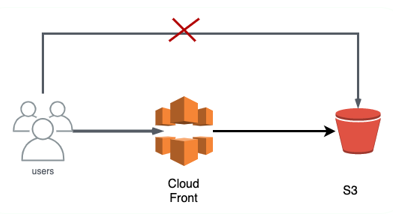
* Signed URLs
* Signed cookies using key pairs Origin Access Identities(OAI)
  + Ensures that only CloudFront can access S3
  + Allow access to S3 only to a special CloudFront user

**Amazon CloudFront - Signed URLs and Cookies**



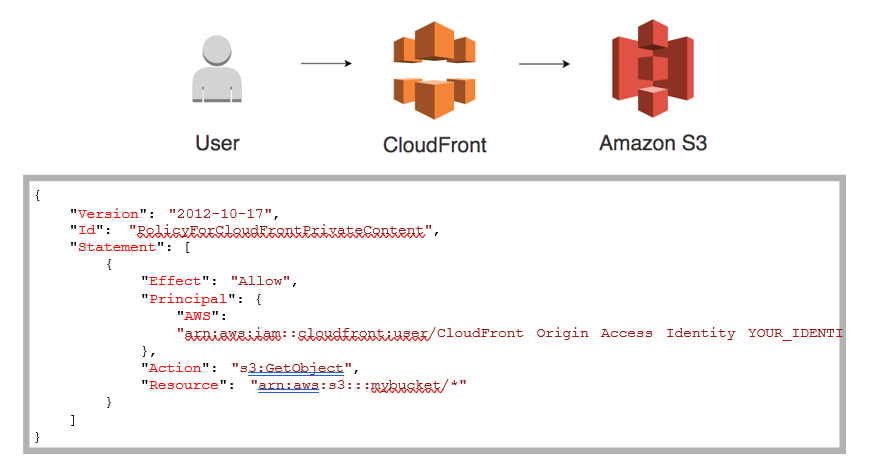
* Signed URLS
  + RTMP distribution
  + Application downloads (individual files) and
  + Situations where cookies are not supported
* Signed Cookies
  + Multiple files (You have a subscriber website)
  + Does not need any change in application URLs

**Amazon CloudFront - Origin Access Identities(OAI)**



* Only CloudFront can access S3
* Create a Special CloudFront user - Origin Access Identities(OAI)
* Associate OAI with CloudFront distribution
* Create a S3 Bucket Policy allowing access to OAI

**Bucket Policy - S3 ONLY through Cloud Front**



**Amazon CloudFront - Remember**

* Old content automatically expires from CloudFront
* Invalidation API - remove object from cache
  + REMEMBER : Designed for use in emergencies
* Best Practice - Use versioning in object path name
  + Example : /images/profile.png?version=1
  + Prevents the need to invalidated content
* Do not use CloudFront for
  + all requests from single location
  + all requests from corporate VPN
* Scenario: Restrict content to users in certain countries
  + Enable CloudFront Geo restriction
  + Configure Whitelist(countries to be allowed) and Blacklist(countries to be blocked)

