AWS CLOUDFRONT

The role of CDNs in a distributed cloud system

- Distributed systems introduce latency
- Global audience accessing static resources
  - Sometimes in high volume!
  - Offload traffic
- CDNs push content to edge locations
  - Giant cache
  - Better performance
  - High availability
- Smart routing to closest location

About Amazon CloudFront

- **Route users to content in edge locations**
  - Reduces the number of network hops
- **Core concepts**
  - Objects
  - Origin server
  - Distribution
  - Edge location
  - Expiration
  - Invalidation
- **Eventual consistency in action**
- **Private distributions supported**

Amazon CloudFront

CloudFront is a Content Delivery Network (CDN) used for distributing cached static files from EDGE locations around the world. Elastic and scalability principles of CloudFront include:

- Easily grow the number of items in a CloudFront distributions that are being served by using Amazon S3 as an origin
- AWS EDGE locations are designed to handle increased connections automatically based off of demand
- CloudFront uses multiple layers of caching on EDGE locations to reduce the load on origin servers such as EC2 instances. This will allow for accepting a growing number of incoming connections without having to scale backend servers.

CloudFront Key Concepts And Overview

Dynamic Content & Whole Site CDN – CloudFront is not just a “static files only” CDN anymore. When you enable “forward query strings” these will now be forwarded to the origin (if the origin supports it S3 does not) which allows the CDN to cache static pages such as word press posts that pull from a database. We’ll learn in whole site CDN how to configure this to ensure if the dynamic content changes it doesn’t stay cached.

Media Streaming – CloudFront allows you to stream media on-demand, Adobe RTMP streaming distributions as well as streaming origins such as WOWZA EC2 instances.

HTTP Methods: Core benefits are allowing you to use CloudFront for all website actions

DELETE – no caching
GET - caching
HEAD - caching
OPTIONS - caching
PATCH – no caching
POST – no caching
PUT – no caching

What does this mean?

1. If you upload an option using PUT it is not cached on the origin even though the upload process uses the closest origin. The origin acts only as a proxy back to AWS which does in fact reduce latency and speeds up the upload process.
2. Delete request will delete the object but not remove it from cache, invalidating the cache is still required.

Dynamic Content With CloudFront

- Use one CDN for an entire website rather than one for just static files.
- Use custom origins and origin rules to determine what part of the website requests go to an origin. For example, images to go S3 but dynamic content goes to a specific EC2 instance.
- Whole site CDN works with uploads as well, up to 20GB. The edge location acts as a proxy for the uploaded object to the origin with the speed of an AWS backed network rather than open internet. This will increase site performance even with uploads!

- Use 0 TTL for dynamic content

Dynamic Content With CloudFront

Scenario: BCJC is consulting for a company that runs their current application entirely all on-premise. However, they are expecting a big boost in traffic tomorrow and need to figure out a way to decrease the load in order to handle the scale. Unfortunately, they cannot migrate their application to AWS in the time period required. What could they do to their current on-premise application to help offload some of the traffic and scale to meet the demand expected in 24 hours?

Whole site CDN! CloudFront allows you to specify custom origins including on-premise servers and sources. Configure static resources in the CDN as well as dynamic content and enable query string forwarding.

Dynamic Content With CloudFront

How does dynamic caching work? What if the dynamic content has changed?!

- Create a custom origin for your dynamic content
- Enable forwarding of query strings
- Set the TTL to 0 IMPORTANT! What does a TTL of 0 do?
  - It will cache the content even though the TTL is set to zero
  - When a request is made it will make a GET request to the origin with an "**If-Modified-Since**" header to determine if there is new data in the origin if there is then the new data is requested and cached else the current data is served from the origin

Dynamic Content With CloudFront

Device Detection: Send users different content based on the type of device that makes the request to the CloudFront origin which is based off of the User Agent header.

Geo Targeting: Serve content specific to an individual country by using CloudFront Geo targeting; URL stays the same content sent is different.

How it works: Essentially AWS now records this information and sends it as part of the request. Your code on the application server can process the data and return customized content based off of the information.

Query Strings / URL parameter forwarding

i.E http://domain.com/videodownloads?current=5 (forwards current = 5 to the server)

CloudFront Reporting

Access Logs: Shows details of every request made to your CloudFront origin. Can integrate with EMR for log analysis.

Log data:

- Object requested
- Date and time of request
- Edge location serving the request
- Client IP address
- HTTP Referrer
- HTTP User Agent

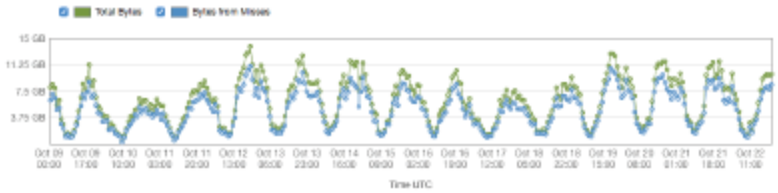
Access logs are sent to and stored in Amazon S3 buckets

CloudFront Reporting: Cache Statistics

Total Requests (Millions | Thousands | Not Scaled) Show Details

Average: 12.6405 K **Total:** 4,208.301 K **Maximum:** 27.94 K **Minimum:** 0.735 K

Bytes Transferred to Viewers (Gigabytes | Megabytes | Kilobytes) Show Details



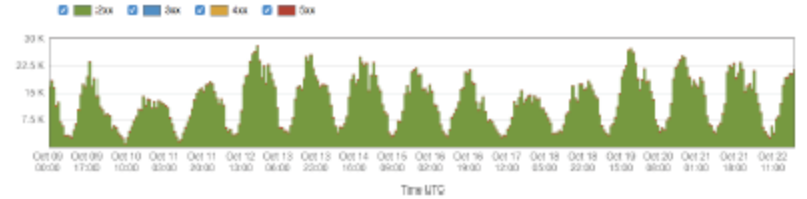
Total Bytes: 2,044.736 GB **Total Bytes from Misses:** 1,642.2974 GB

Percentage of Viewer Requests by Result Type Show Details

Percentage of GET Requests that Didn't Finish Downloading Show Details

Total: 1.11%, 46.68 K of 4.21 M **Minimum:** 0%, 2 **Maximum:** 0.01%, 427

HTTP Status Codes (Millions | Thousands | Not Scaled) Show Details



Total 2xx: 4,192.006 K (99.99%) **Total 3xx:** 0 K (0%) **Total 4xx:** 0.32 K (0.01%) **Total 5xx:** 0.052 K (0%)

QUALITY THOUGHT
INFOSYSTEMS (INDIA) PVT. LTD.

CloudFront Reporting: Additional Reports & Analytics

Popular Objects: Shows the most requested objects from the CDN distribution

Top Referrers: Shows the URL that made the most requests to the CDN distribution

Usage: Number of HTTP / HTTPS requests, Data transferred By Protocol, Data Transferred By Destination (From CloudFront To The Users / From CloudFront To The Origin)

Viewers:

- Devices
- Browsers
- Operating Systems
- Locations

CloudFront Security

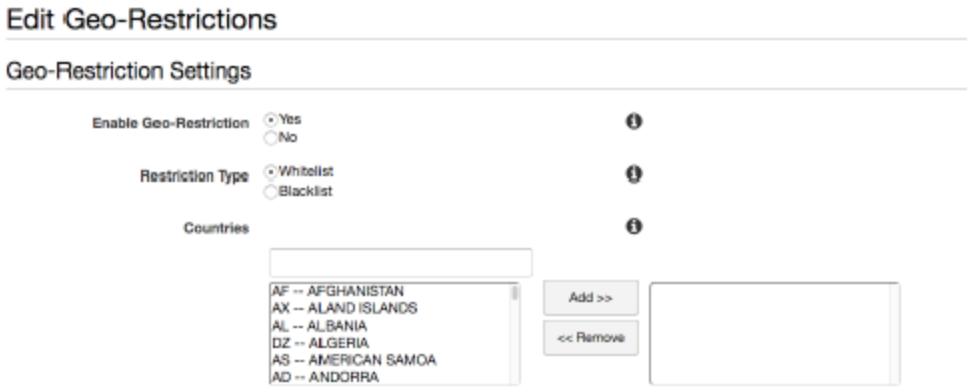
Private Content

- Signed URLs: Provide URLs with expire dates to limit access to content

- Signed Cookies: Signed cookies are new and are an extremely flexible tool in terms of limiting content. You can limit content without limiting access to the URL. For example: if a user is logged into a site, you can issue a signed cookie that verifies they have permission to access certain parts of the site. If streaming HLS files from CloudFront you can also create signed cookies that will be validated each time an HTTP request is made to an HLS chunk. Essentially, providing secure streaming!

CloudFront Security

Geo Restriction: A CloudFront setting that allows you to specify which countries your CDN will deliver to



CloudFront Performance Considerations

- Increase performance by increasing the number of requests that are cache hits instead of cache misses

- Use CloudFront to upload objects, the edge location will proxy the data back to the origin location going over the AWS backend network

- Increase minimum TTL and maximum TTL so items are cached longer (if they are not frequently changing)

How does Cloud Front React in the event of high load and multiple simultaneous requests?

In case of increase in simultaneous requests CloudFront, will wait for the first request to finish before processing the second request.