Agenda

- **Building a React app..**
  - **Development versus Production environment.**
  - **Dynamic versus Static versus of a React app.**

- **Frontend app deployment to AWS platform.**
  - **What services?**
    - **Ans: S3 and CloudFront.**

- **Fullstack web app.**
  - **Deployment as a single CDK app**
    - **Dynamic Integration**
    - **Custom domain.**

Amazon CloudFront CDN

CloudFront / CDN

- **A CDN (Content Delivery Network) uses a network of geographically dispersed servers (edge locations or POPs) to cache copies of content close to end users, lowering latency when they download or stream objects.**

- **CloudFront – 400+ Points of Presence (POP) in 90 cities and across 47 different countries.**

- DDoS protection.

Cloudfront – Resources.

1. **Origin – The source of your content, e.g. S3.**
2. **Distribution - A distribution tells CloudFront where you want content to be delivered from, and the details about how to track and manage content delivery.**

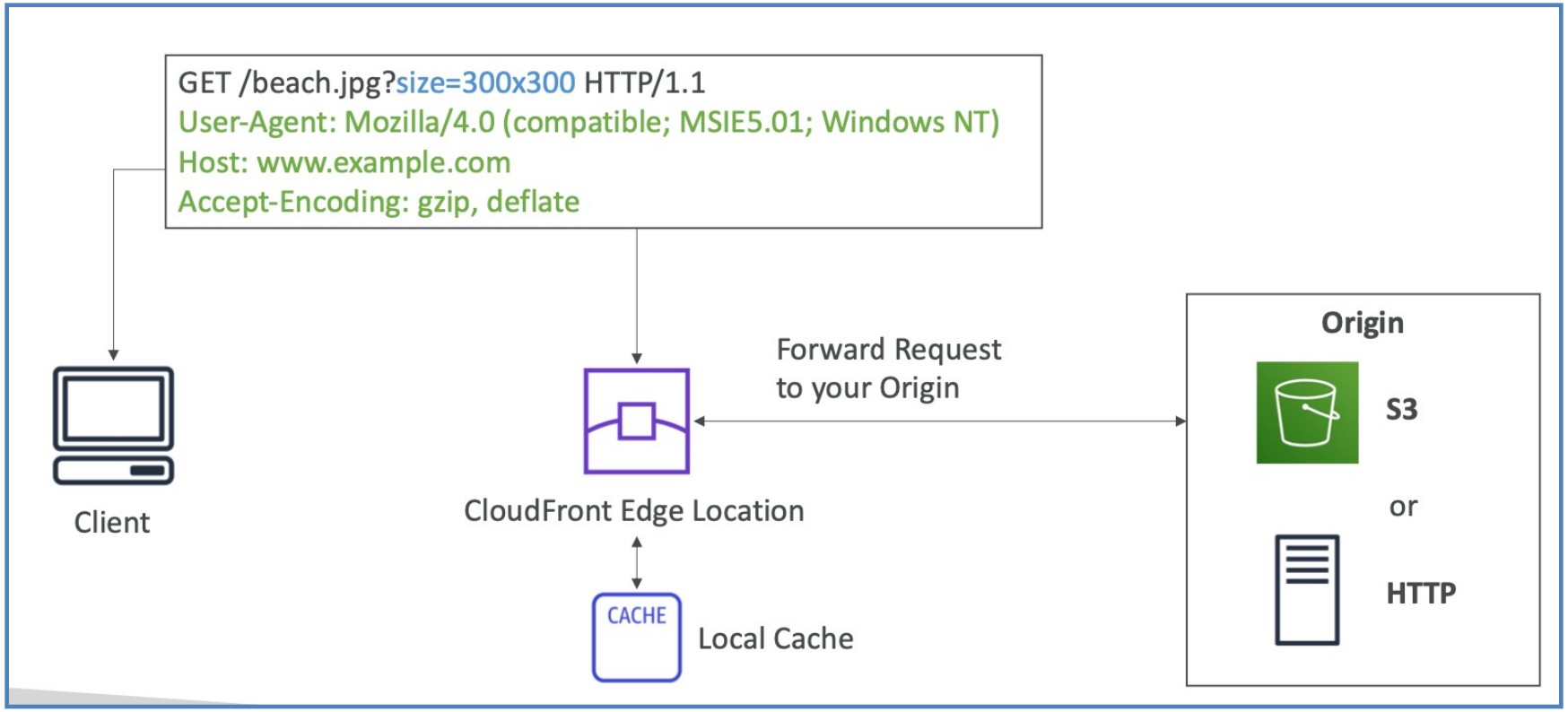
Cloudfront - Origin

- **Source of your content.**

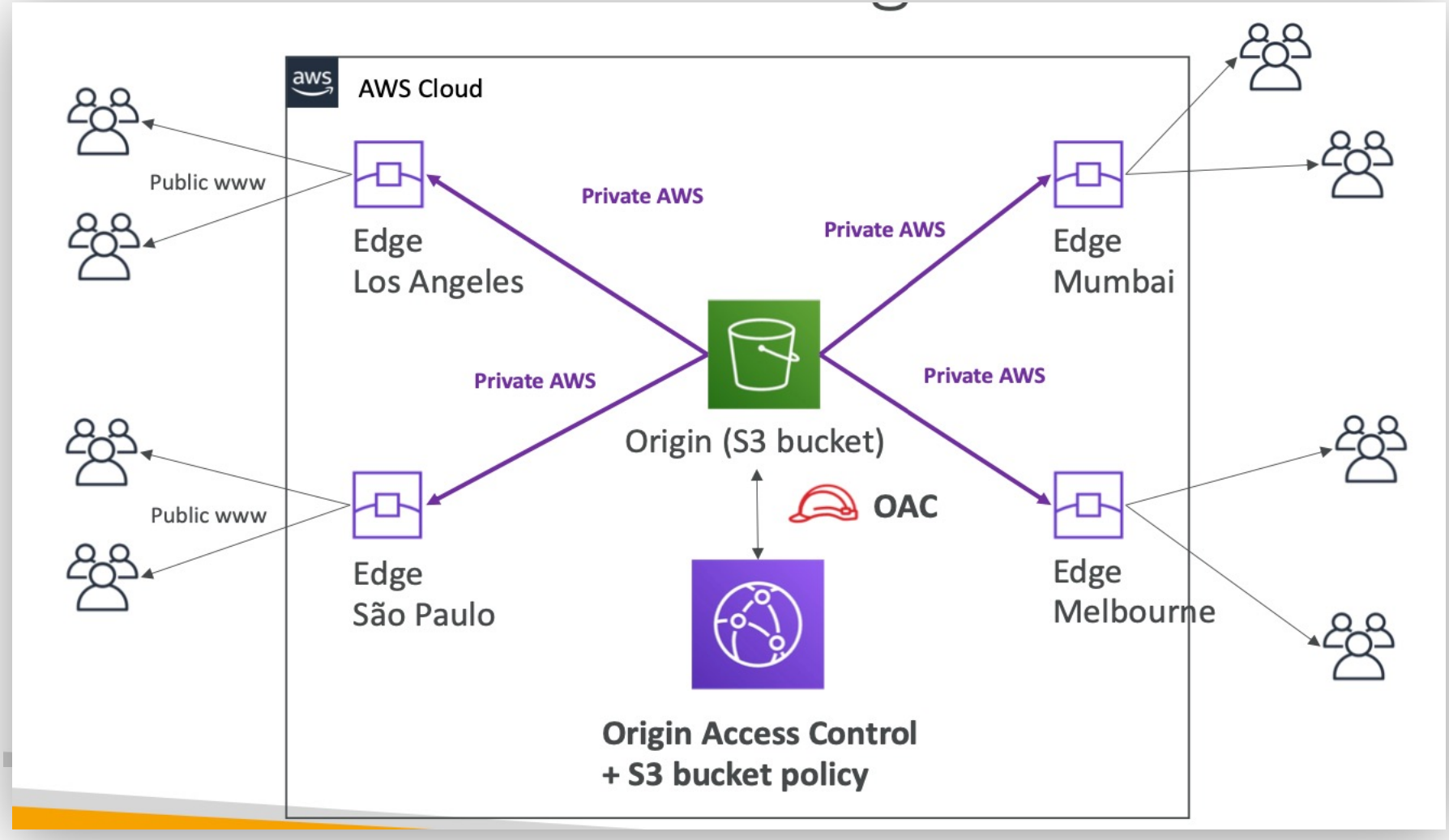
1. **S3 bucket.**
   - **For distributing files and caching them at the edge**
   - **Enhanced security with CloudFront Origin Access Identity (OAI)**
     - **Origin Access Control (OAC) Is replacing OAI.**
   - **CloudFront can be used as an *ingress* (to upload files to S3)**

1. **Custom origin (HTTP).**
   - **API Gateway.**
   - **Application load Balancer.**
   - **EC2 instance.**
   - **S3 website.**

CloudFront – High level view.

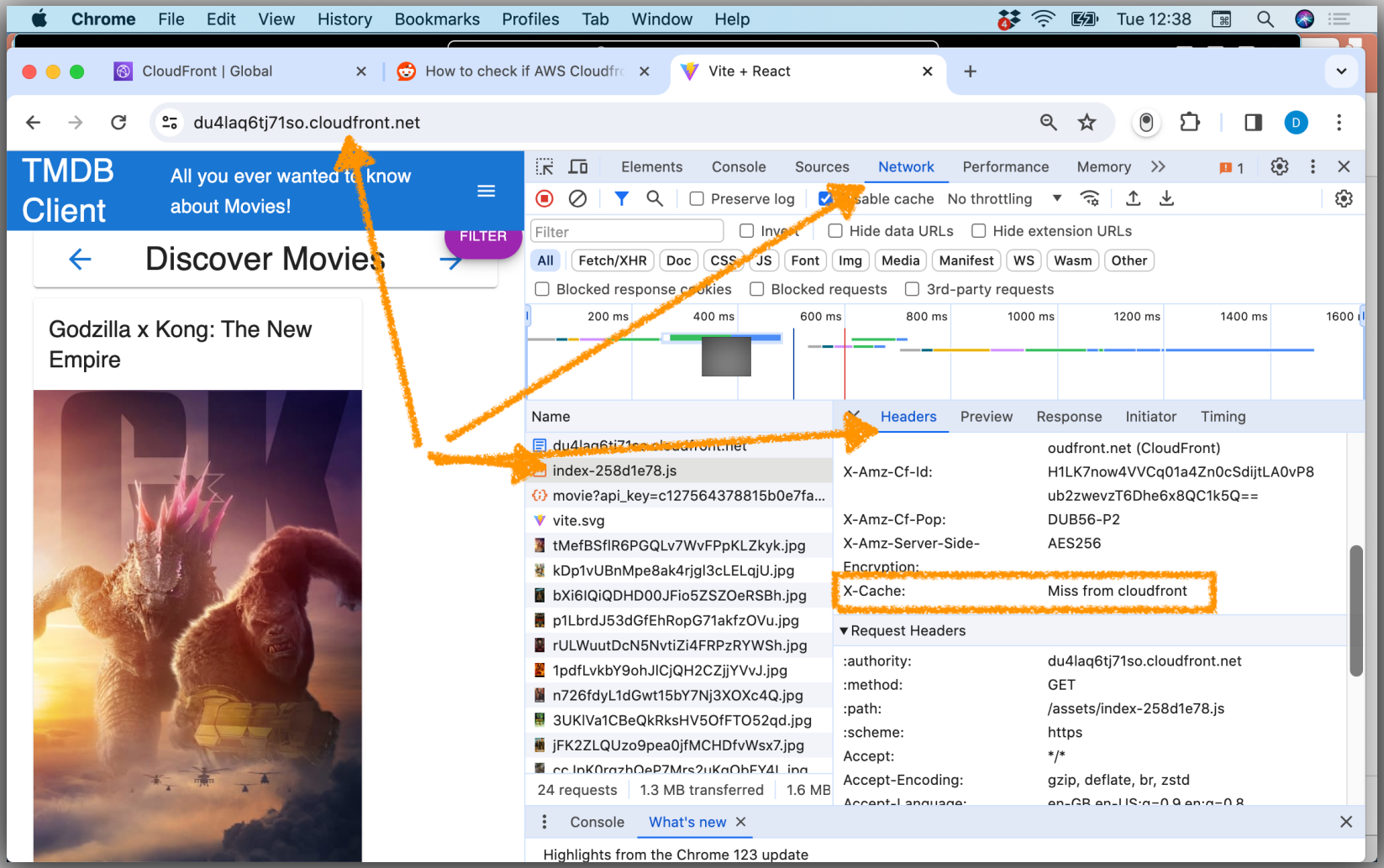


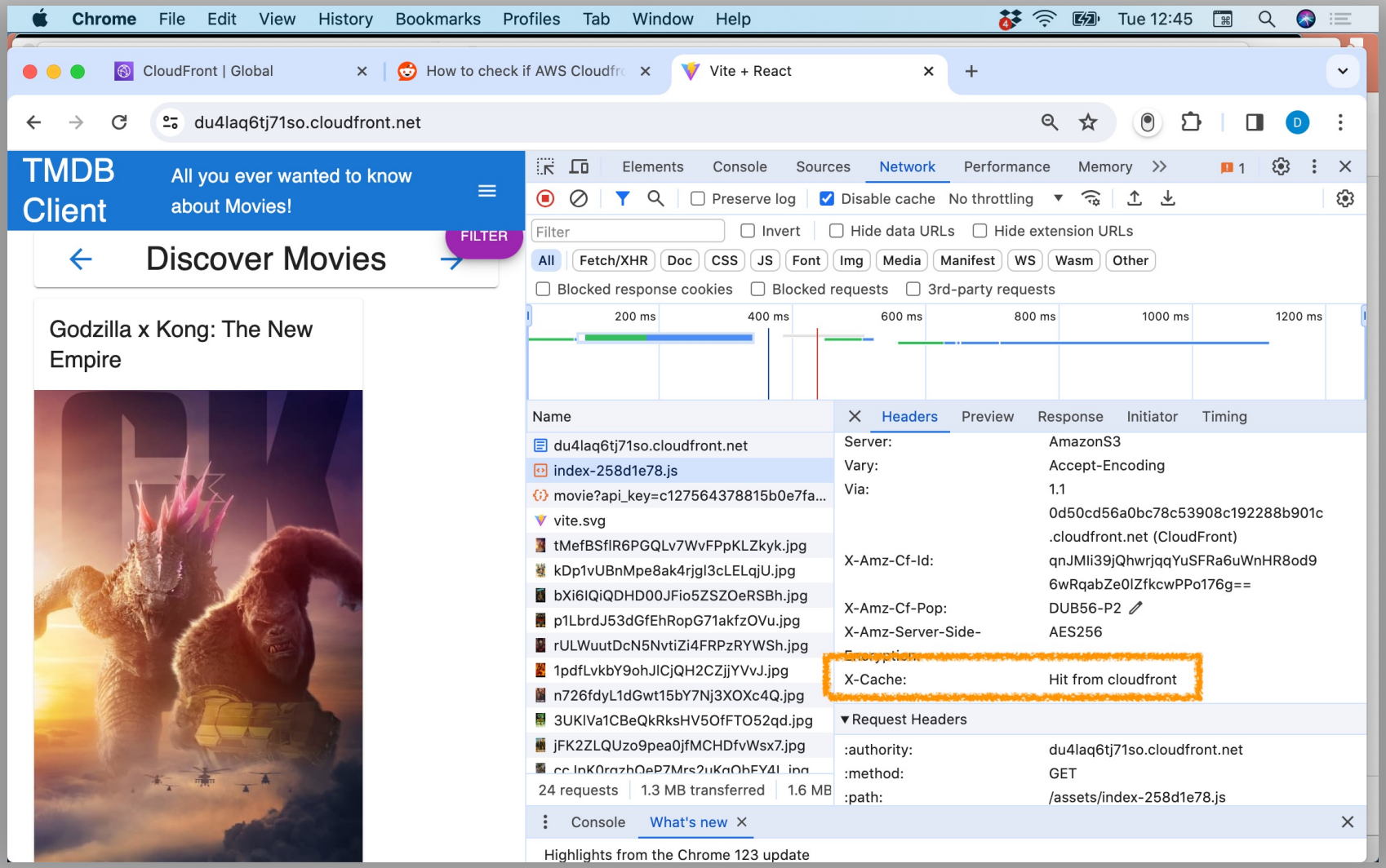
S3 as an Origin



Demo

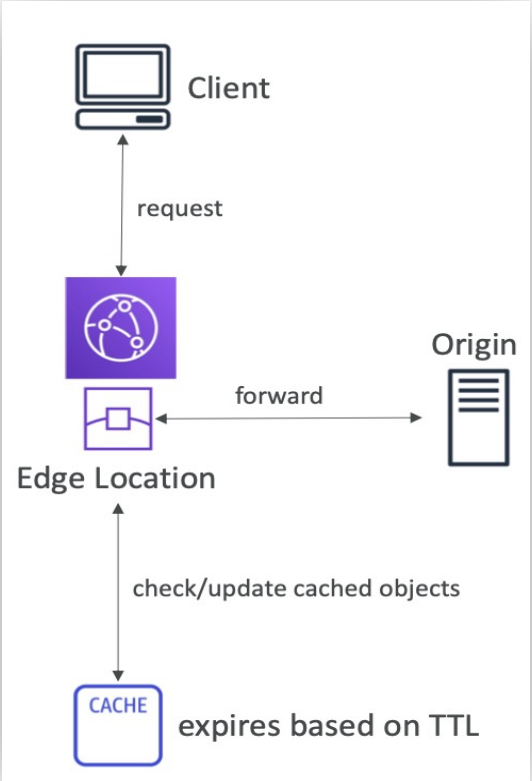
- **Objective: Provision a Cloudfront distribution for the Movies React app, using S3 as the origin.**


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Tue 12:38
CloudFront | Global
How to check if AWS Cloudfr
Vite + React
du4laq6tj71so.cloudfront.net
TMDB Client
All you ever wanted to know about Movies!
FILTER
Discover Movies
Godzilla x Kong: The New Empire
Elements Console Sources Network Performance Memory
Preserve log
No throttling
Filter
Hide data URLs
Hide extension URLs
All Fetch/XHR Doc CSS JS Font Img Media Manifest WS Wasm Other
Blocked response cookies
Blocked requests
3rd-party requests
200 ms 400 ms 600 ms 800 ms 1000 ms 1200 ms 1400 ms 1600
Name
index-258d1e78.js
movie?api_key=c127564378815b0e7fa...
vite.svg
tMefBSfIR6PGQLv7WvFPpKLZkyk.jpg
kDp1vUBnMpe8ak4rjgl3cLELqjU.jpg
bXi6IQiQDHD00JFio5ZSZOeRSBh.jpg
p1LbrdJ53dGfEhRopG71akfzOVu.jpg
rULWuutDcN5NvtiZi4FRPzRYWSh.jpg
1pdfLvkbY9ohJlCjQH2CZjjYVvJ.jpg
n726fdyL1dGwt15bY7Nj3XOXc4Q.jpg
3UKlVa1CBeQkRksHV5OfTO52qd.jpg
jFK2ZLQUzo9pea0jfMCHDfvWsx7.jpg
24 requests 1.3 MB transferred 1.6 MB
Headers Preview Response Initiator Timing
oudfront.net (CloudFront)
X-Amz-Cf-Id: H1LK7now4VVCq01a4Zn0cSdijtLA0vP8ub2zwevzT6Dhe6x8QC1k5Q==
X-Amz-Cf-Pop: DUB56-P2
X-Amz-Server-Side-Encryption: AES256
X-Cache: Miss from cloudfront
Request Headers
:authority: du4laq6tj71so.cloudfront.net
:method: GET
:path: /assets/index-258d1e78.js
:scheme: https
Accept: */*
Accept-Encoding: gzip, deflate, br, zstd
Console What's new
Highlights from the Chrome 123 update


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Tue 12:45
CloudFront | Global
How to check if AWS Cloudfr
Vite + React
du4laq6tj71so.cloudfront.net
TMDB Client
All you ever wanted to know about Movies!
FILTER
Discover Movies
Godzilla x Kong: The New Empire
Elements Console Sources Network Performance Memory
Preserve log
Disable cache
No throttling
Filter
Invert
Hide data URLs
Hide extension URLs
All Fetch/XHR Doc CSS JS Font Img Media Manifest WS Wasm Other
Blocked response cookies
Blocked requests
3rd-party requests
200 ms 400 ms 600 ms 800 ms 1000 ms 1200 ms
Name
du4laq6tj71so.cloudfront.net
index-258d1e78.js
movie?api_key=c127564378815b0e7fa...
vite.svg
tMefBSfIR6PGQLv7WvFPpKLZkyk.jpg
kDp1vUBnMpe8ak4rjgl3cLELqjU.jpg
bXi6IQiQDHD00JFio5ZSZOeRSBh.jpg
p1LbrdJ53dGfEhRopG71akfzOVu.jpg
rULWuutDcN5NvtiZi4FRPzRYWSh.jpg
1pdfLvkbY9ohJlCjQH2CZjjYVvJ.jpg
n726fdyL1dGwt15bY7Nj3XOXc4Q.jpg
3UKlVa1CBeQkRksHV5OfFTO52qd.jpg
jFK2ZLQUzo9pea0jfMCHDfvWsx7.jpg
24 requests 1.3 MB transferred 1.6 MB
Headers Preview Response Initiator Timing
Server: AmazonS3
Vary: Accept-Encoding
Via: 1.1 0d50cd56a0bc78c53908c192288b901c.cloudfront.net (CloudFront)
X-Amz-Cf-Id: qnJMli39jQhwrjqqYuSFRa6uWnHR8od96wRqabZe0lZfkcwPPo176g==
X-Amz-Cf-Pop: DUB56-P2
X-Amz-Server-Side-Encryption: AES256
X-Cache: Hit from cloudfront
Request Headers
:authority: du4laq6tj71so.cloudfront.net
:method: GET
:path: /assets/index-258d1e78.js
Console
What's new
Highlights from the Chrome 123 update

CloudFront Cache

- A cache is a high performance key-value store.
- The CF cache lives at each CloudFront Edge Location
- CloudFront identifies each object in the cache using the *Cache Key*
- You want to maximize the Cache Hit ratio to minimize requests to the origin.
- You can *invalidate* part of the cache using the CreateInvalidation API



CloudFront cache key

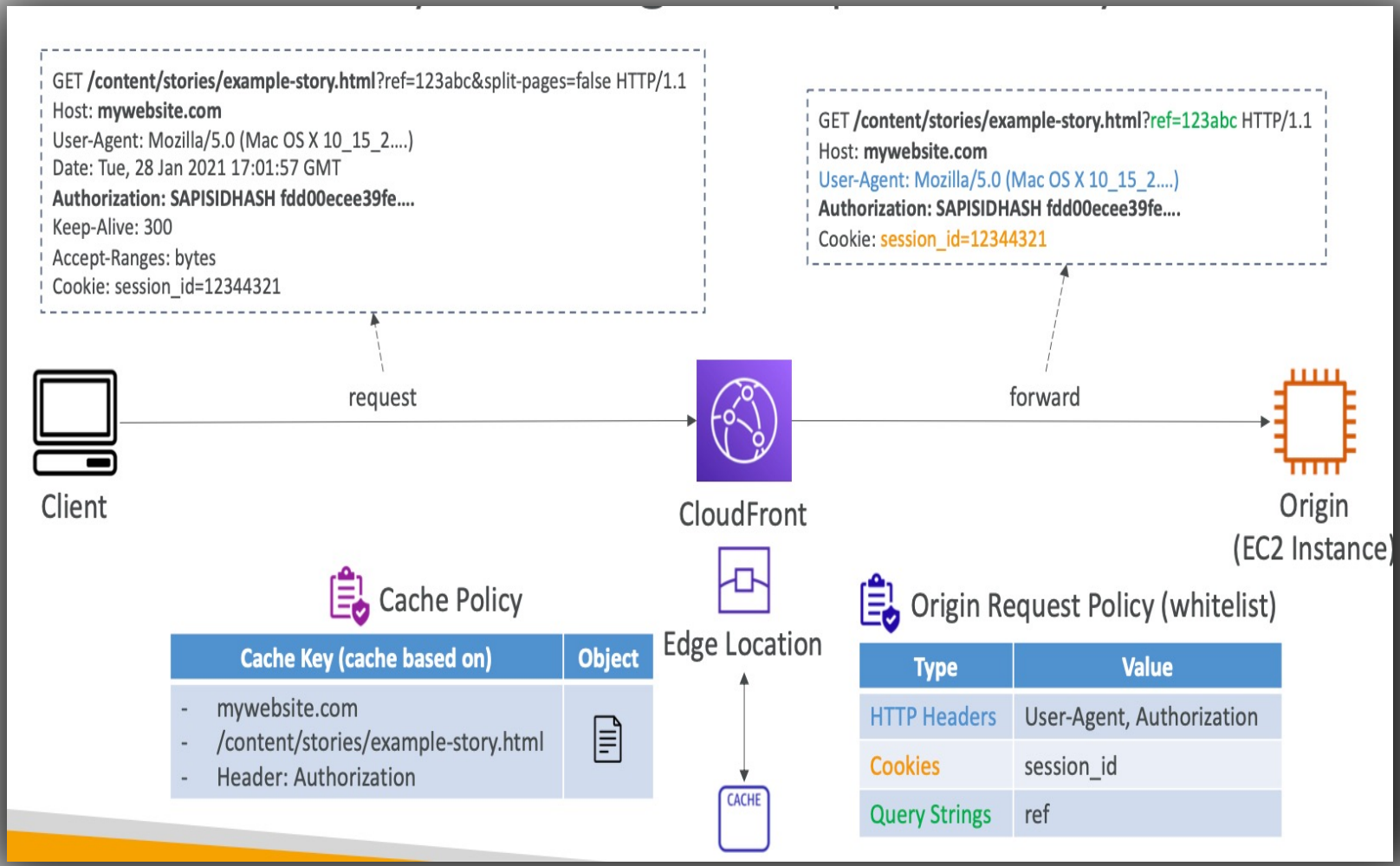
- **Cache key is a unique identifier for every object in the cache store.**

- **(Default) Key = Hostname + Resource portion of URL**
- **E,g, HTTP Get http://donainX.com/blog/article20.html?ref=123ab&split-pages=false → Key = donainX.com/blog/article20.html**

- **If you have an application that serves up content that varies based on user, device, language, location... you can add other elements (HTTP headers, cookies, query strings) to the Cache Key using CloudFront** *Cache Policies*

CloudFront - Policies

1. Cache policy.
   - Key based on:
     - HTTP Headers: None – Whitelist
     - Cookies: None – Whitelist – Include All-Except – All
     - Query Strings: None – Whitelist – Include All-Except – All
   - Control the TTL (0 seconds to 1 year)..
2. Origin Request Policy.
   - Specify values that you want to include in origin requests without including them in the Cache Key (no duplicated cached content)
   - HTTP headers: None – Whitelist – All
   - Cookies: None – Whitelist – All
   - Query Strings: None – Whitelist – All


GET /content/stories/example-story.html?ref=123abc&split-pages=false HTTP/1.1
Host: mywebsite.com
User-Agent: Mozilla/5.0 (Mac OS X 10_15_2....)
Date: Tue, 28 Jan 2021 17:01:57 GMT
Authorization: SAPISIDHASH fdd00ecee39fe....
Keep-Alive: 300
Accept-Ranges: bytes
Cookie: session_id=12344321
GET /content/stories/example-story.html?ref=123abc HTTP/1.1
Host: mywebsite.com
User-Agent: Mozilla/5.0 (Mac OS X 10_15_2....)
Authorization: SAPISIDHASH fdd00ecee39fe....
Cookie: session_id=12344321
request
forward
Client
CloudFront
Origin
(EC2 Instance)
Edge Location
CACHE
Cache Policy
Cache Key (cache based on)
Object
- mywebsite.com
- /content/stories/example-story.html
- Header: Authorization
Origin Request Policy (whitelist)
Type
Value
HTTP Headers
User-Agent, Authorization
Cookies
session_id
Query Strings
ref

Cache Invalidation

- In case you update the back-end origin, CloudFront doesn't know about it and will only get the refreshed content after the TTL has expired.
- However, you can force an entire or partial cache refresh (thus bypassing the TTL) by performing a CloudFront Invalidation
- You can invalidate all files (*) or a special path (/images/*)

Fullstack Web App.

- Objective Use the CDK framework to provision the cloud infrastructure for a Fullstack web app.
    - How does the frontend resolve the backend API(s) URL?
        1. Manually – Hardcoded.
        2. Automatically - Dynamically at deployment time.

Custom Domain Name

- Objective Use the CDK framework to allow a custom domain name be linked to a Cloudfront-enamled web app
- Steps:

1. Buy a domain name - see https://porkbun.com/ - Watch - https://www.youtube.com/watch?v=kl3a76CBwX4
2. In the AWS Route53 console, create a Hosted Zone for your domain name.
3. **Copy the hosted zone's list of name servers to your registered domain name on its Domain Registry (e.g. porkbun).**
4. To support HTTPS traffic, create a Certificate for your domain/subdomain using the AWS Certificate Manager service in the us-east-1 region.
5. Attach the certificate to your Cloudfront distribution for the particular domain/subdomain.
6. Add an A record to your hosted zone to redirect requests to your app's custom URL to the CF distribution.

Referencs.

- https://github.com/diarmuidoconnor/cdk-s3-simple-website
- https://github.com/diarmuidoconnor/cdk-s3-cloudfront-route53