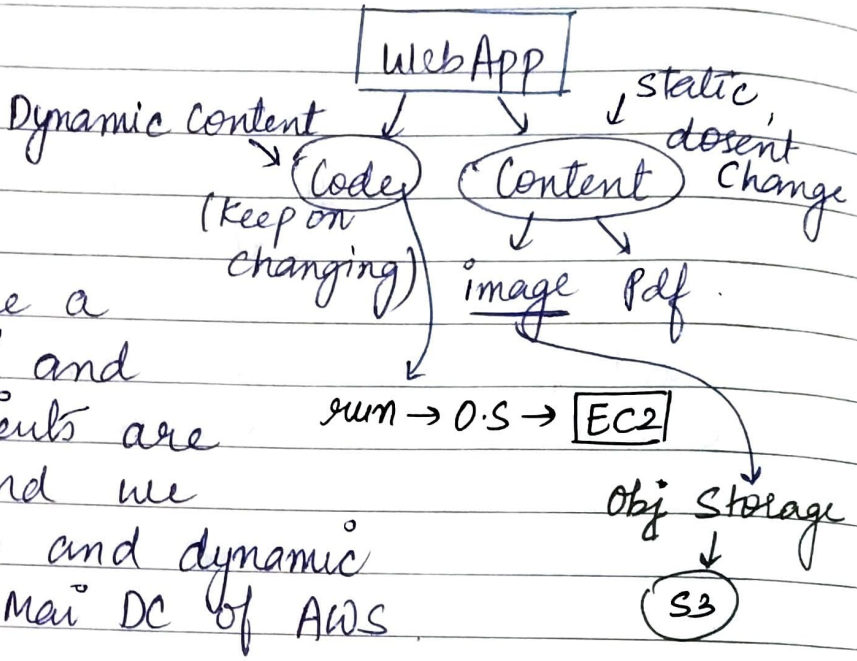
DAY-5

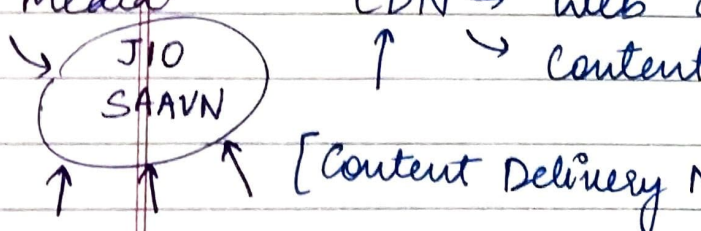
▷ Cloud front

Low Latency Content Delivery



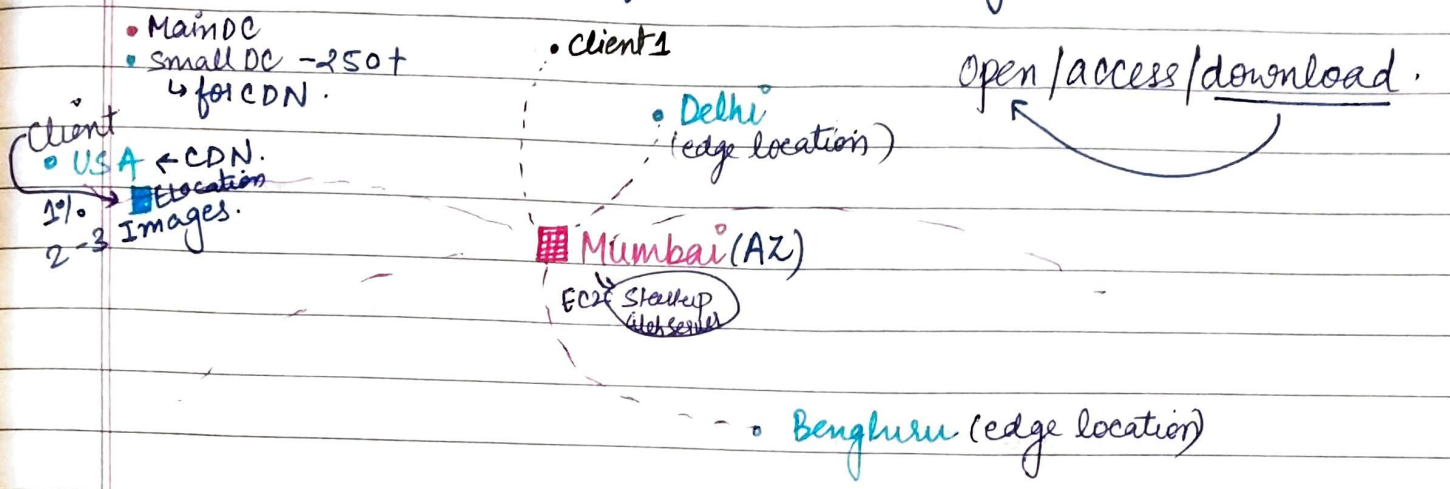
→ Suppose we have a new startup 'A' and our initial clients are in Mumbai and we put our static and dynamic content in MuMai DC of AWS.

Suppose your startup expands and you have customes from USA therefore your USA client needs to access data from Mumbai DC that will result in latency and what they may not like.
Your client would like a Quick response And your content would be delivered - through Network



↳ Object Storage as a Service (S3)
↳ Block Storage as a Service (EBS)

What can we do for low-latency ?



* Edge location - to deliver the content.
- the mostly frequent data / content that client of delhi (other cities) access are kept in the CDN, therefore resulting in low latency.

→ Client need not to travel to main AZ for the content they can access their nearest edge location.

[Client never travels to main DC.
↳ Suppose a client in USA want to access an image that is not present in nearest edge location therefore, on the clients behalf the aws edge location travels. and fetches the data from Mumbai DC comes back to the client nearest-edge location and make a copy of it and then delivers the content to the USA client.
↳ Proxy process.

local copy
↓
Cache.

find
not find

▷ Work of Edge location :-
Cache our content for a copy to edge location to achieve latency.
TTL - Time to Live
- The time the image / content resides in the edge location.
- We can set our TTL (by default - 1 day)

⇒ Content will exist in a service and that will act as Origin of that Content

AWS
└ CDN ⟶ Cloud Front

One unique url for a particular content / website and through that same url everybody around world will access the content.

Distribute Web Content → CDN.
↓
AWS Cloudfront

→ automatically connects to edge 1 or nearest edge location → fault tolerance

- For fault tolerance in EC2 we go for ELB. [Elastic Load Balancer]

→ JioSaavan uses AWS CloudFront for streaming

▷ Practical

- Origin S3 ← Object in S3 / content. [Bucket] ← nobody can access this bucket/content directly [restrict bucket]
- Suppose some image - we will have an image url as soon as we put it in S3.
- CDN → Cloudfront (global) ↳ edge → URL ← Client
  - first time we create CDN it will take 15-20 minutes time as its globally created → (Create Distribution)
- speed-up content delivery through → http/https protocol (static & dynamic content)
- but for live streaming → RTMP (protocol) Real Time ...

Restrict Bucket Access

→ data can be only accessed through CDN and not S3 url directly

for S3 only wants CDN service to access data then we have to create

↓

Bucket Policy.

CDN → Grant Read Permission on Bucket ✓
(Inside Create Distribution)

↓

⊙ Yes, update bucket policy.

S3 ↑ Origin — Edge location ↑ CDN. — Client ↑ Viewer.

- https : we have encryption
  http : not secure.

http://google.co.in → redirects itself

↓

https://google.co.in.

Origin access Identity - like a username

CloudFront → Popular Objects

↓

Miss / Hits can be seen.

→ Reports and Analytics

↳ see upon users / viewers.

Edit Geo - restrictions

↳ We can blacklist countries to deny them access to your website.