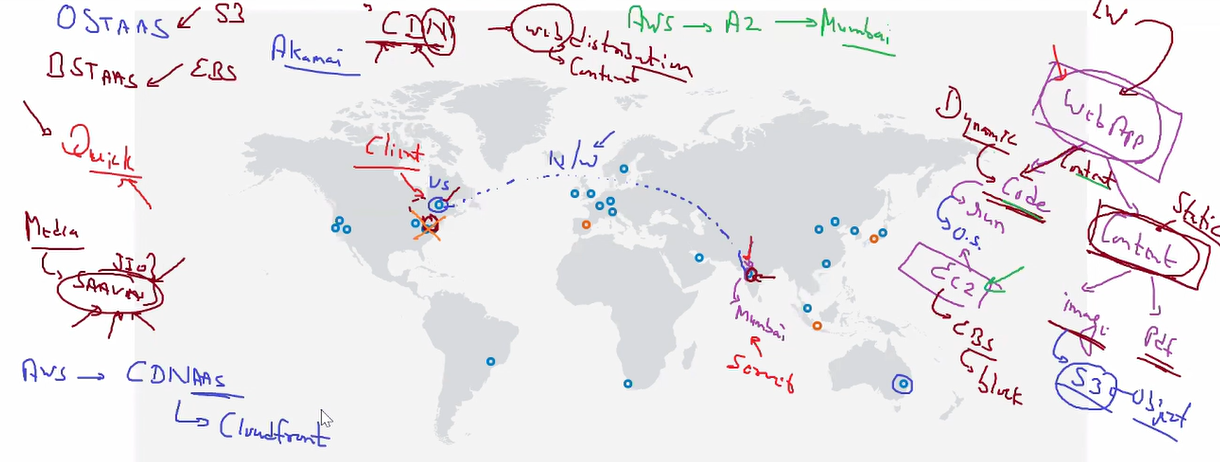
Cloud Front

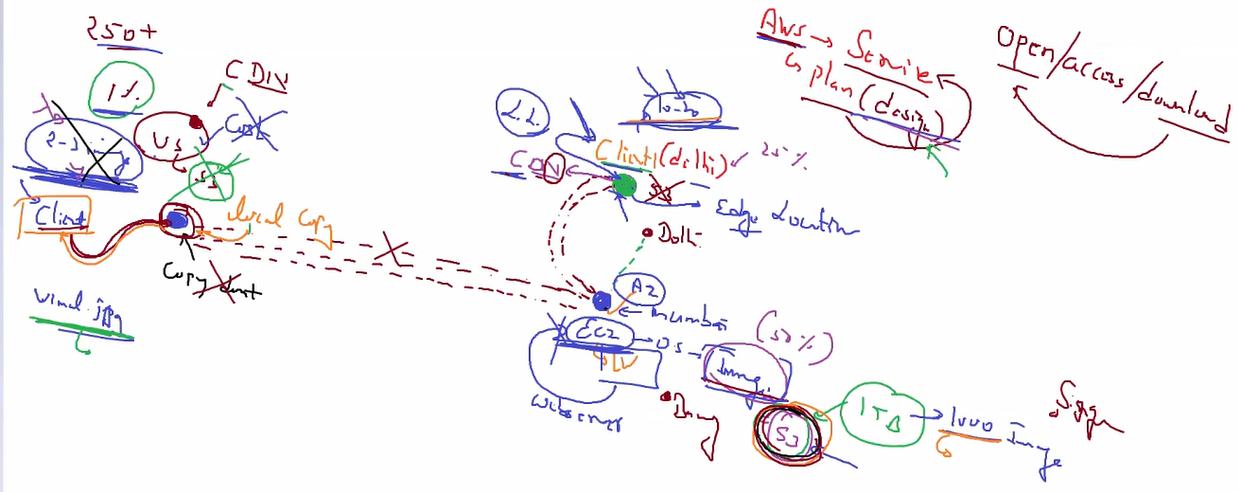
* Mainly our website - Webapp contains 2 type of things.

1. Code
2. Content
   1. Image audio video

* For static content we use S3.
* Suppose You hosted your website in Mumbai and client is located in Mumbai (50%), delhi (25%), US (1%).
  + Now because most of your client belongs to Mumbai you have to host take s3 bucket in Mumbai.
  + But when you access a web-app code contains only some kb code. But your images and videos are in MBs and GBs.
  + So if US client connect to web-app it will take him a longer duration to access a web-app.
    - It will increase your latency.

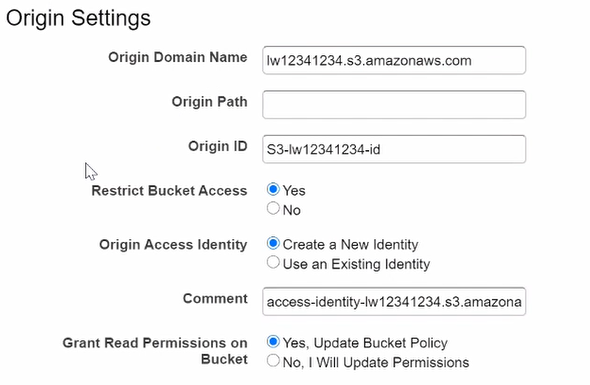


* For this type of service, we have content delivery type of services.
  + Many companies provide CDN services, one of famous one is Akamai.
  + AWS also provides you CDN services.
  + CDN is a concept name, for this aws has one service called CloudFront.
* Here what we do, we have multiple small data centers in multiple locations.
  + These small data centers are known as Edge Locations.
  + They are just meant for storing the content (Image, Audio, Video).
  + So suppose US client want to access a Web-app then code will come from Mumbai but we will store content in US nearby edge location so content has to travel little bit distance.
  + And as you click, complete web-app is come up including images and code without any downtime.
    - AWS has 250+ Edge location across the world.
* So, for this we have to use CloudFront, lets see how can we use it.
  + If we deploy complete S3 storage to edge location, it will lead you to very huge amount for production.
    - As you know we use S3 URL in web-app (code).
  + So, what we do is, when any client search for web-app in US, code goes to US from Mumbai.
  + CloudFront provides you one URL.
  + Now this URL is very intelligent. It will draw your request to the nearest data center, it will also follow fault tolerance, If any edge location is down, it will lead to 2nd nearest data center.
  + We can use this URL in our main code (instead of S3 url) and we can get content from nearest location.
* How is the overall scenario?

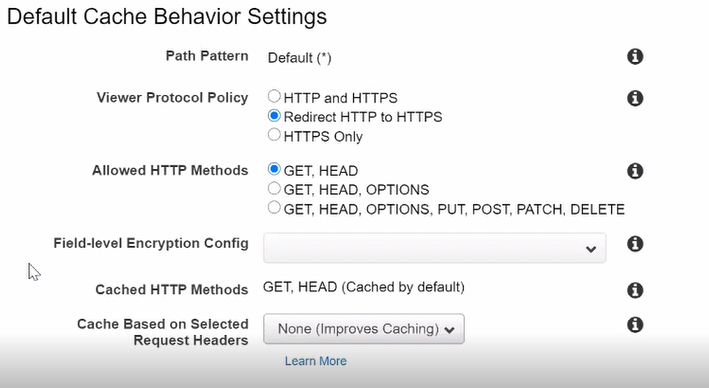


1. US client wants to access Mumbai Web-app.
2. His request goes to Mumbai and server returns the code.
3. Code contains CloudFront URL.
4. URL leads to nearest edge location and get data from there.
   1. When client hits first URL has not any data.
   2. So on behalf of client this URL go to Mumbai S3 and retrieve data from there.
      1. Only those data which clients want.
      2. We may have 30GB data in S3, but client require only 5 images, so we will retrieve only those 5 images and store it in the edge location as local copy.
         1. Same location people have same mindset.
         2. So mostly they also search for same few images.
      3. These copies we also send to client.
   3. Now when 2nd client comes to same Edge Location it will get data within a second because we already have that data present there.
5. This URL returns the image and we can access complete site.
   1. First client may have to wait for some seconds so amazon can create local copy in the edge location.

* Some important keywords
  + Proxy
    - Instead of client URL goes to S3 origin and create local copy.
  + Miss
    - URL lead to edge location & it don’t have content.
  + Hit
    - URL lead to edge location and it has content.
  + TTL
    - For how much edge location has content available.
    - Default 1 day (86400 seconds)
      * Min – 0
      * Max – 31536000 – 1 year
  + OAI – Origin Access Identity
    - Unique identity for accessing Origin (S3).
  + Less latency
  + Fault tolerance
  + Single URL
    - Intelligent
* Demo
  + Create S3 bucket with image.
    - Block publicly access.
      * We can access content only using CloudFront.
  + Create CloudFront
    - Origin Domain name – where is content available.
      * S3 bucket.
    - Origin Path – which file/folder has code.
    - Origin ID – Just a unique ID
    - Restrict Bucket Access – Yes
    - Origin Access identity – Create new identity
      * According to CloudFront S3 is origin. It would like to access (Only to him) to S3 bucket. I am giving my (unique) identity.
    - Grant Read permission on bucket – Yes, update policy.
      * We have to Update S3 that only CloudFront can access.
      * For this we have to create policy.
      * By this policy is automatically created.

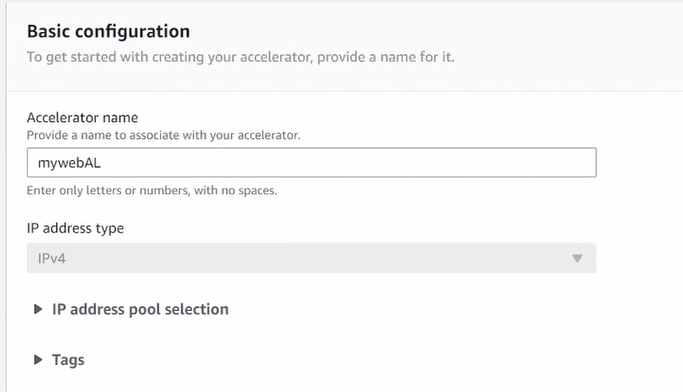
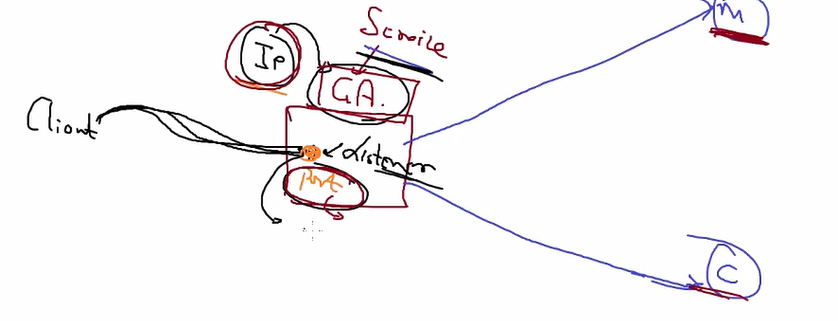
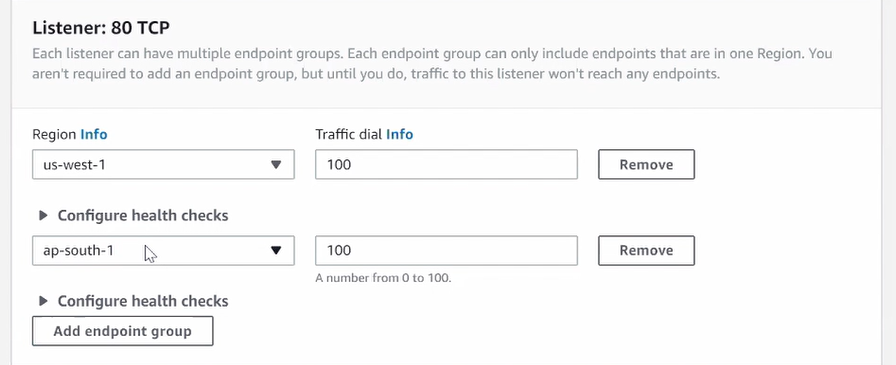
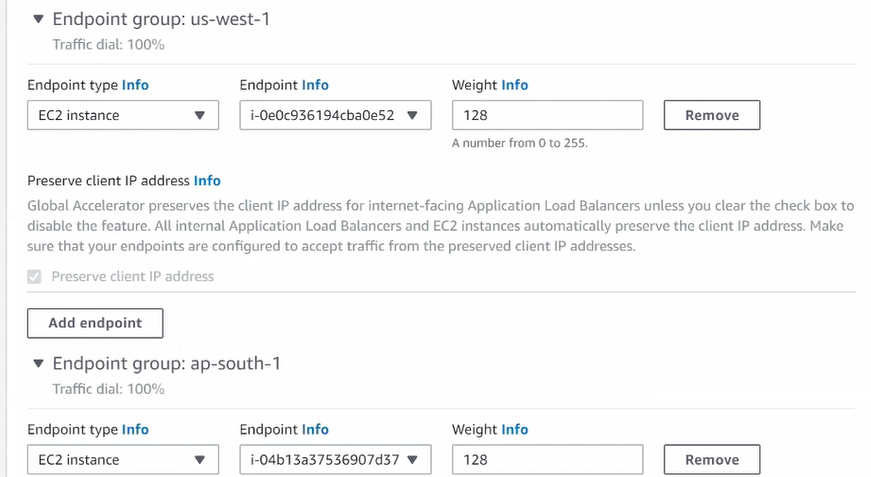


* + - Viewer protocol policy – redirect http to https
      * If any client connect to CF URL using http it will be automatically converted into https.



* + - Price class – Use All Edge location.
      * Here you can mention which edge location you want to use.
    - Distribution State – enabled.
    - Create distribution
  + It will take some time to create a unique URL and upload it into all edge location.
  + Click on you link.
    - You can restrict the viewers geo graphically also.
    - Whitelist
      * Allow all selected countries.
    - Blacklist
      * Block all selected countries.
  + Add this URL into your web-app.
* You are ready to go.
* Instead of audio/ Video you can also use edge location for storing complete web-app.
* Now we have created website also in the CloudFront with default TTL 86400.
* But suppose your web-app is very critical and you have changed something in the web-app.
  + But cache will not update till 1 day.
  + So here we can use invalidation.
    - It will clear cache for all edge location.

Global Accelerator

* Due to some compliance issue, national policy or some other reason we cannot copy data in multiple location.
  + We only host few things in edge location, we don’t store everything there.
  + So we have to reach the other location.
  + If we use public internet then we get more latency and less security.
* So here we can use private global network from AWS.
  + We have to go to edge location and from edge location we can reach to anywhere using global network of amazon.
  + For this we have to use service called global accelerator.
* For check this search global accelerator speed test in google.
* Suppose our most of client from US and India so we have launch our web-app at both location.
  + If US client goes to US DC but somehow our web-app is down in US, so they will redirect to Mumbai DC using global accelerator.
  + 
  + Listener – Client connect to URL using which port.
  + 
  + 
  + 

Cloud Watch

* It is a monitoring service for aws cloud resources.
* It can also collect metrics and logs from resources.
* Two types of monitoring
  + Basic monitoring
  + Detailed Monitoring
* It will allow you to record metrics for EBS, EC2, ELB, S3.
* We can set some events based on cloud watch.
  + Eg, trigger AWS lambda
* Concerned With **what happening with AWS resources**.
* You can use Cloud watch Logs to see what is going on your website.

Cloud trail

* Concerned with **Who did what on AWS**.
* enables governance, compliance, operational auditing, and risk auditing.
* Who is doing what in AWS account.

Bastion OS

* Launch public – private subnet setup.
* For best practices in private subnet instance SG only allow Private subnet SG.
* Create two route table.
  + Public subnet is attached to public-rt.
    - 0.0.0.0/0 🡪 Igw
  + Private subnet is attached to private-rt.
    - 0.0.0.0/0 🡪 NAT
    - NAT gateway is launched in one of the public subnets.
* NAT gateway only allow to private instances to go to internet.
  + Private Instance 🡪 Public Instance 🡪 Outside world.
* We can SSH to public instance and using that go to private instance using amazon private network.
  + This OS is known as bastion host.