AWS Global Infrastructure

Introduction

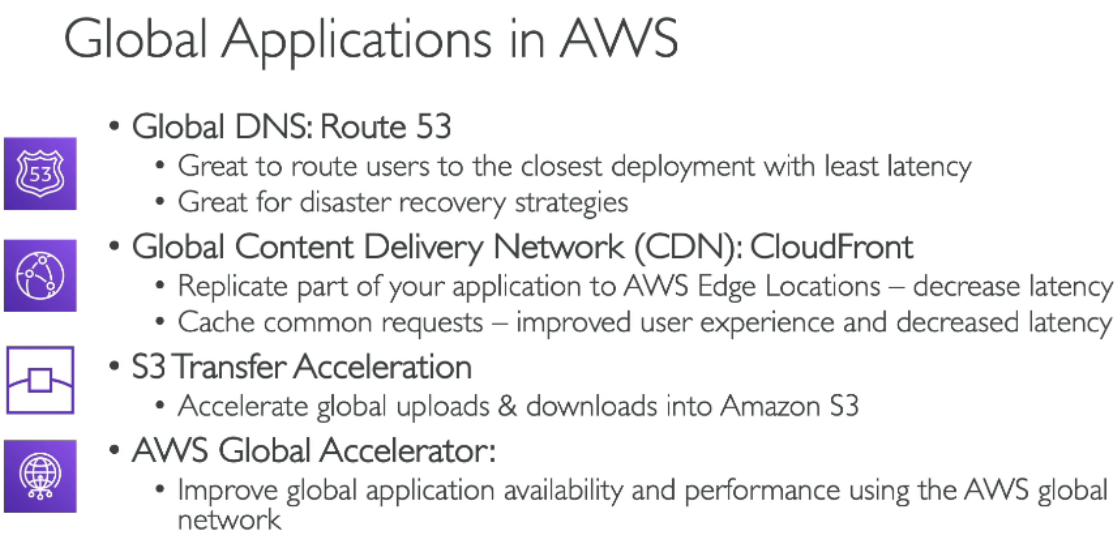
- Why would you need a global application?

  Deploying your application globally, in multiple geographies through AWS (by deploying in multiple regions and edge locations) could have several benefits like:

  1. Decreased Latency
  2. Smoother user experience
  3. Disaster Recovery
  4. Attack protection - distributed global infrastructure is harder to attack all at once.

- AWS' Global Infrastructure overview
  - You have **regions** where you deploy your applications and infrastructure. A region consists of multiple AZs.
  - **AZs or Availability Zones** are made up of multiple data centres.
  - **Edge locations or Points of Presence (PoP)** are essential for content delivery as close as possible to the users.

- To Learn:



Route 53

- AWS Route 53 is a scalable and highly available Domain Name System (DNS) web service.
- It helps connect human-readable web addresses (like www.example.com) to the actual locations of services or resources on the internet.
- Scalable, reliable, offers seamless integration with other AWS services.

Key features:

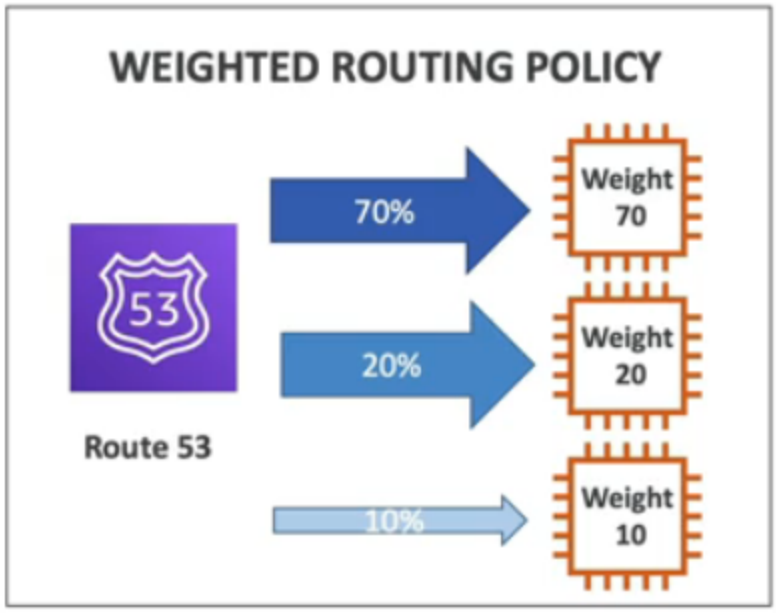
- **Domain registration**
  - Route 53 allows you to register domain names (like www.example.com).
  - Once you register a domain, you own the right to use that specific web address.
- **DNS Management:**
  - You can manage the DNS records associated with your domain. DNS records are like instructions that tell the DNS system how to direct the

traffic coming to your domain.

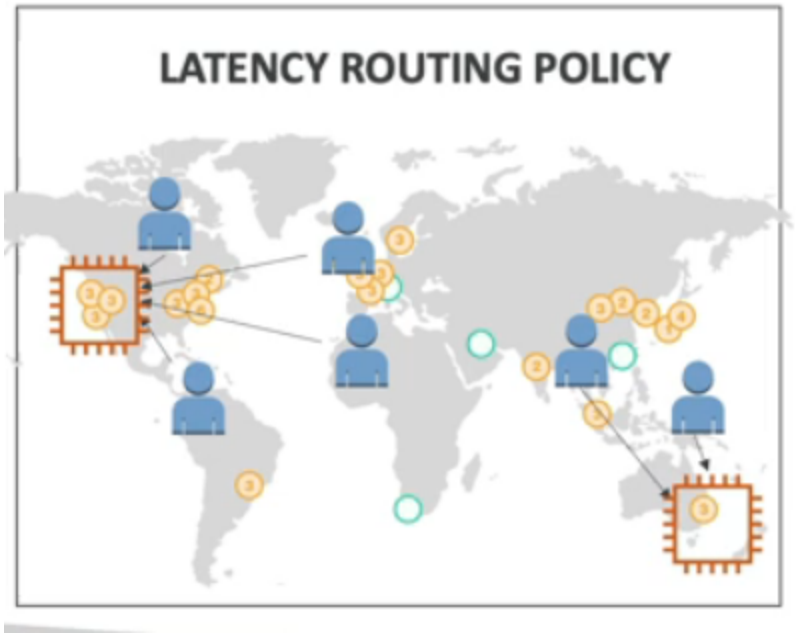
  - Common types of DNS records include:
    - A Record: Maps a domain to an IPv4 address.
    - AAAA Record: Maps a domain to an IPv6 address.
    - CNAME Record: Alias of one domain to another.
    - Alias Record: Maps domain to an AWS resource.
    - MX Record: Specifies mail servers for the domain.

- **Load Balancing:**
  - If you have multiple servers hosting your website, Route 53 can distribute the incoming traffic among them.
- **Health Checks:**
  - Route 53 allows you to monitor the health of your resources associated with the DNS like web servers, etc. If it detects a problem, it automatically redirects traffic to healthy resources.
- **Global Anycast Routing:**
  - This feature helps improve the speed and availability of your application by routing end-user requests to the nearest AWS data center.
- **Routing Policies (imp for CP exam):**
  - You can define how Route 53 should route traffic to your resources using different policies.
  - There are different types of routing policies:

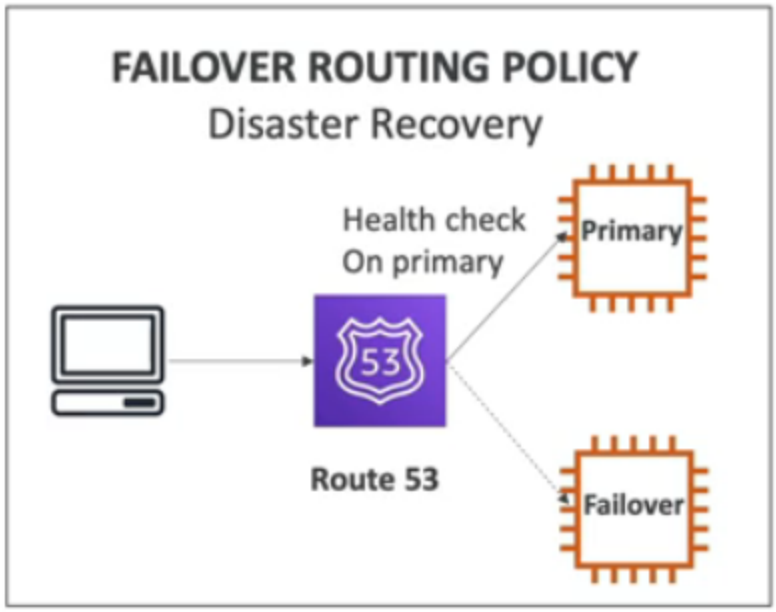
1. **Simple routing policy** - no health checks - web browser goes into DNS system and gets the IP for the domain. Only this policy has no health checks.
2. **Weighted routing policy** - allows to distributed the incoming traffic to different EC2 instances based on weight assigned.



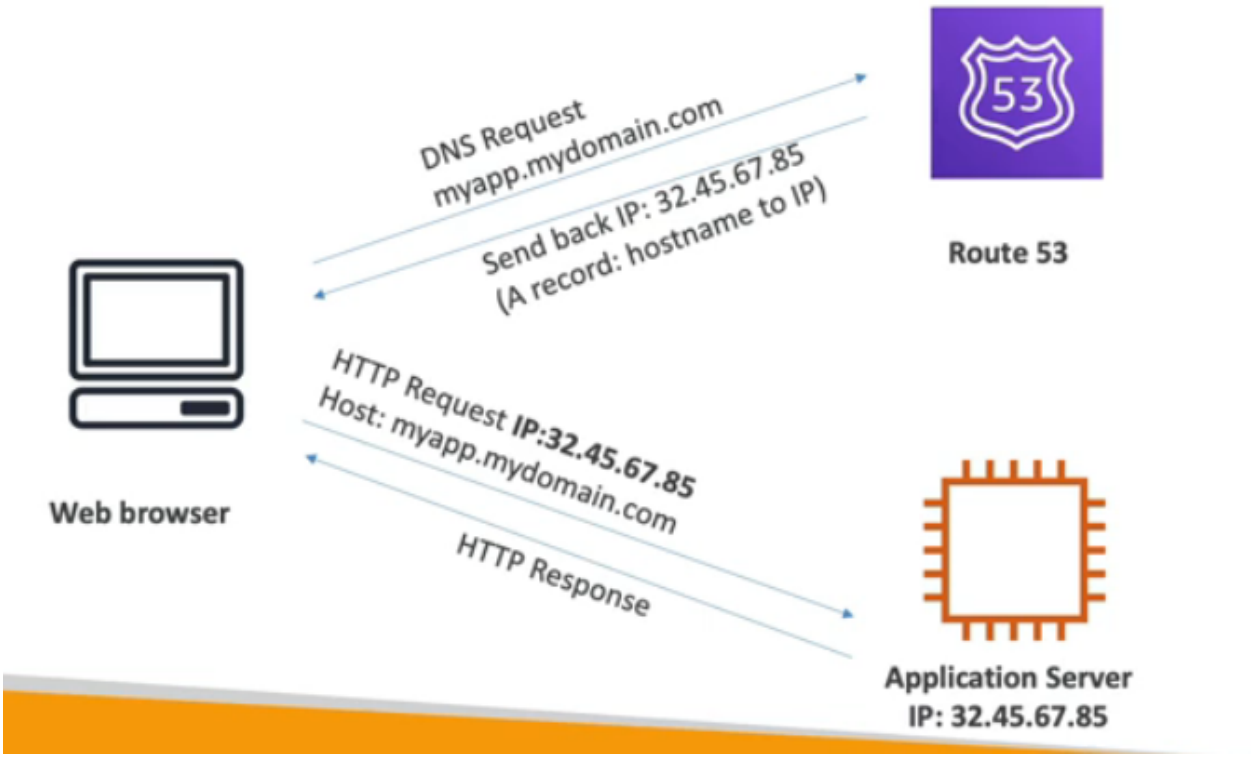
3. **Latency routing policy** - minimizes latency by making the users connect to the servers that is the closest to them.



4. **Failover routing policy** - if on health checks, the primary server comes out as unhealthy, route 53 will redirect the traffic to the failover server.



How Route 53 works?

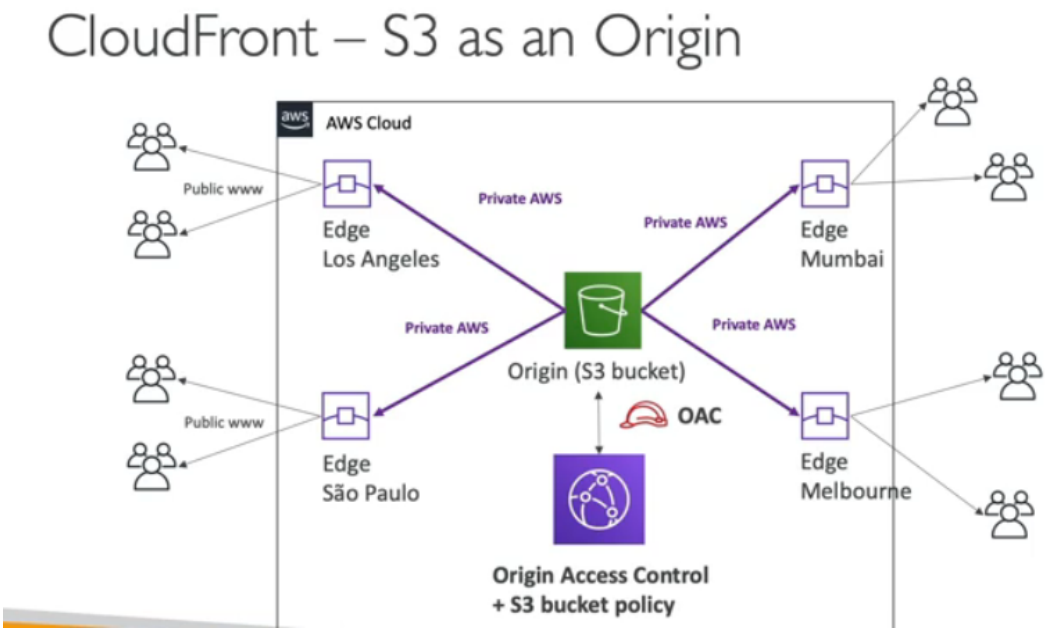


AWS CloudFront

- AWS CloudFront is a content delivery service / network (CDN).
- Its primary goal is to deliver web content (like images, videos, scripts, and other files) to users with low latency and high transfer speeds.
- It improves read performance by caching the content of your webapp at several edge locations around the world which leads to low latency and better user experience.
- In simpler terms, CloudFront helps make websites and applications load faster for users around the world.
- CloudFront is associated with 216 Points of Presence globally.
- Using CloudFront allows you to protect your applications from DDoS attacks due to the servers having global presence and integration with more security aws services like Shield and Web App Firewall.

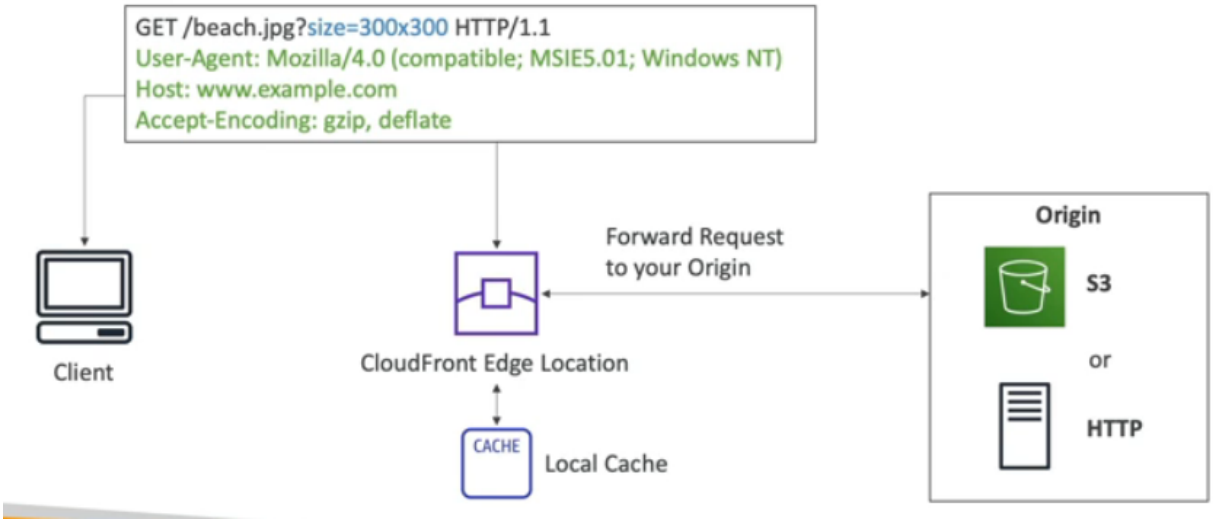
CloudFront Origins

1. S3 Bucket
    - For distributing files and caching them at the edge
    - If you want only your CloudFront to access the S3 bucket then you can enabled something called OAC (Origin Access Control).
    - CloudFront can also be used as an **ingress** (to upload files into a S3 bucket)



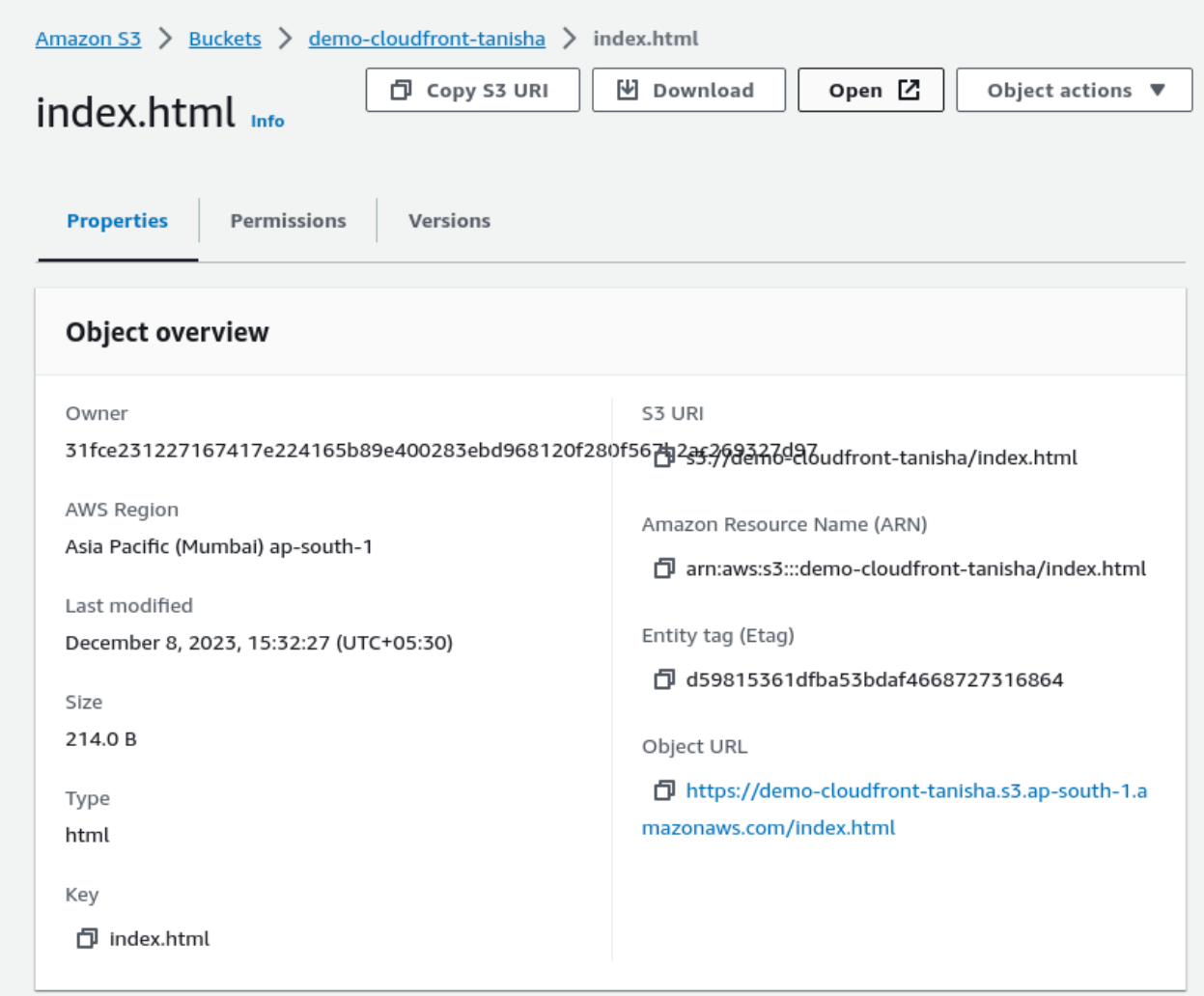
2. Custom Origin (HTTP)
    - You can have CloudFront in front of any custom origin HTTP backend.
    - Including:
        - application load balancer - It could be your website running on an application load balancer, which helps manage and distribute traffic to different servers.
        - ec2 instance - It could be on a specific computer server (EC2 instance) that hosts your website.
        - s3 hosted website - If you have a website stored on Amazon S3, CloudFront can work with it, but you need to set up the S3 bucket as a static website.
        - Basically, CloudFront can work with any website or application that uses the HTTP protocol.
    - This feature allows you to use CloudFront to speed up the delivery of your website or application, regardless of where you store your content, as long as it uses the HTTP protocol.

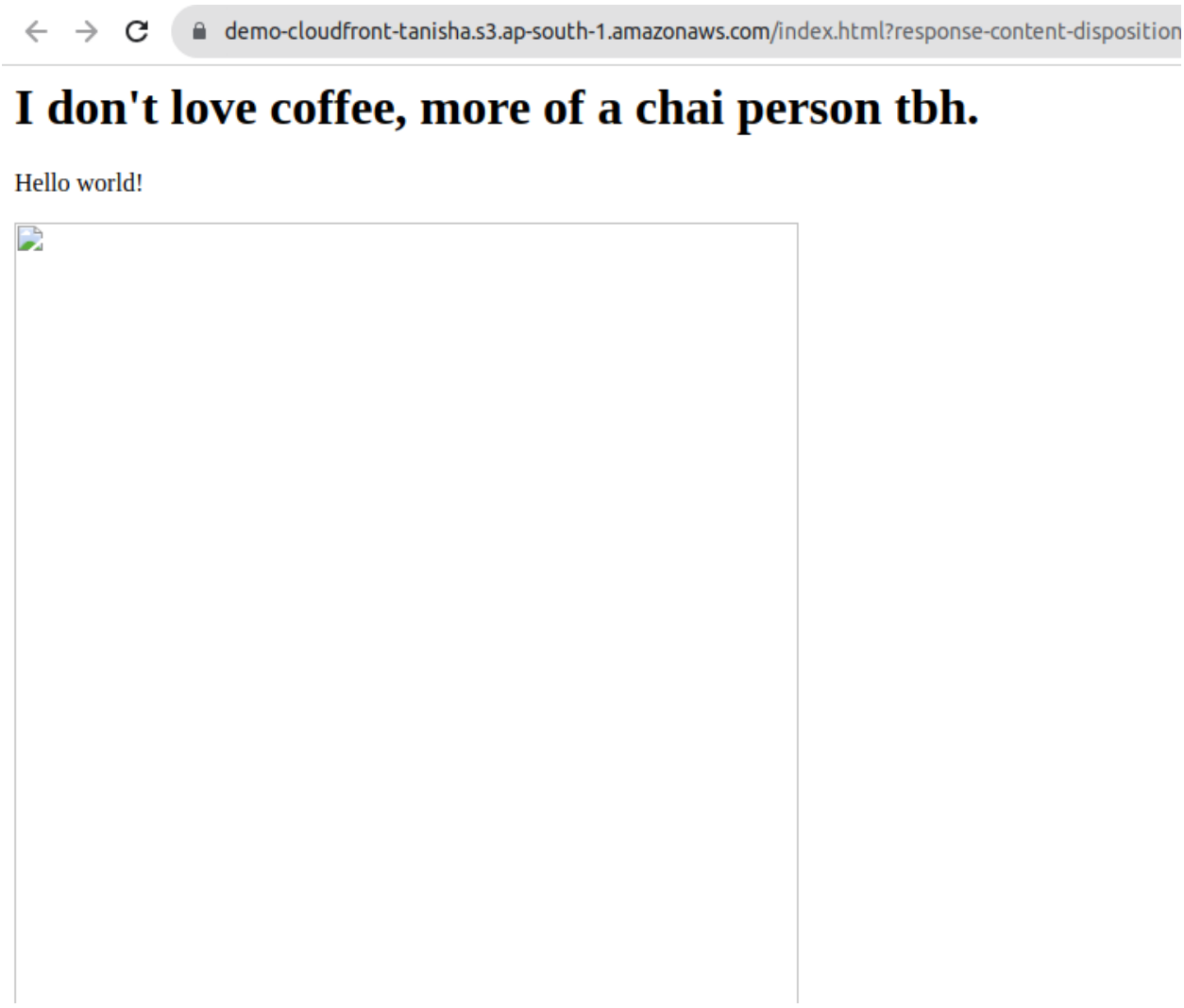
How does CloudFront work?



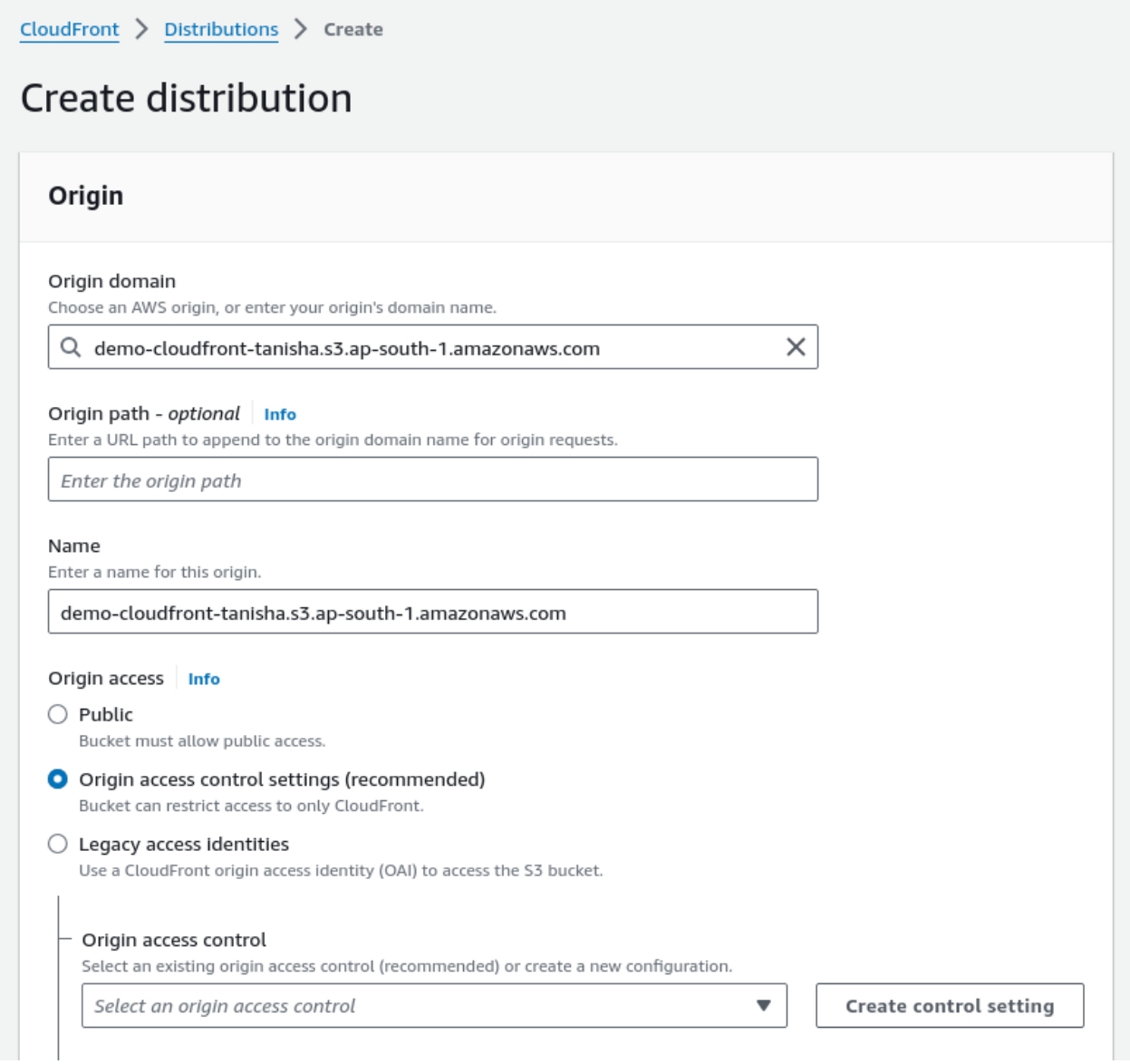
CloudFront Hands-on

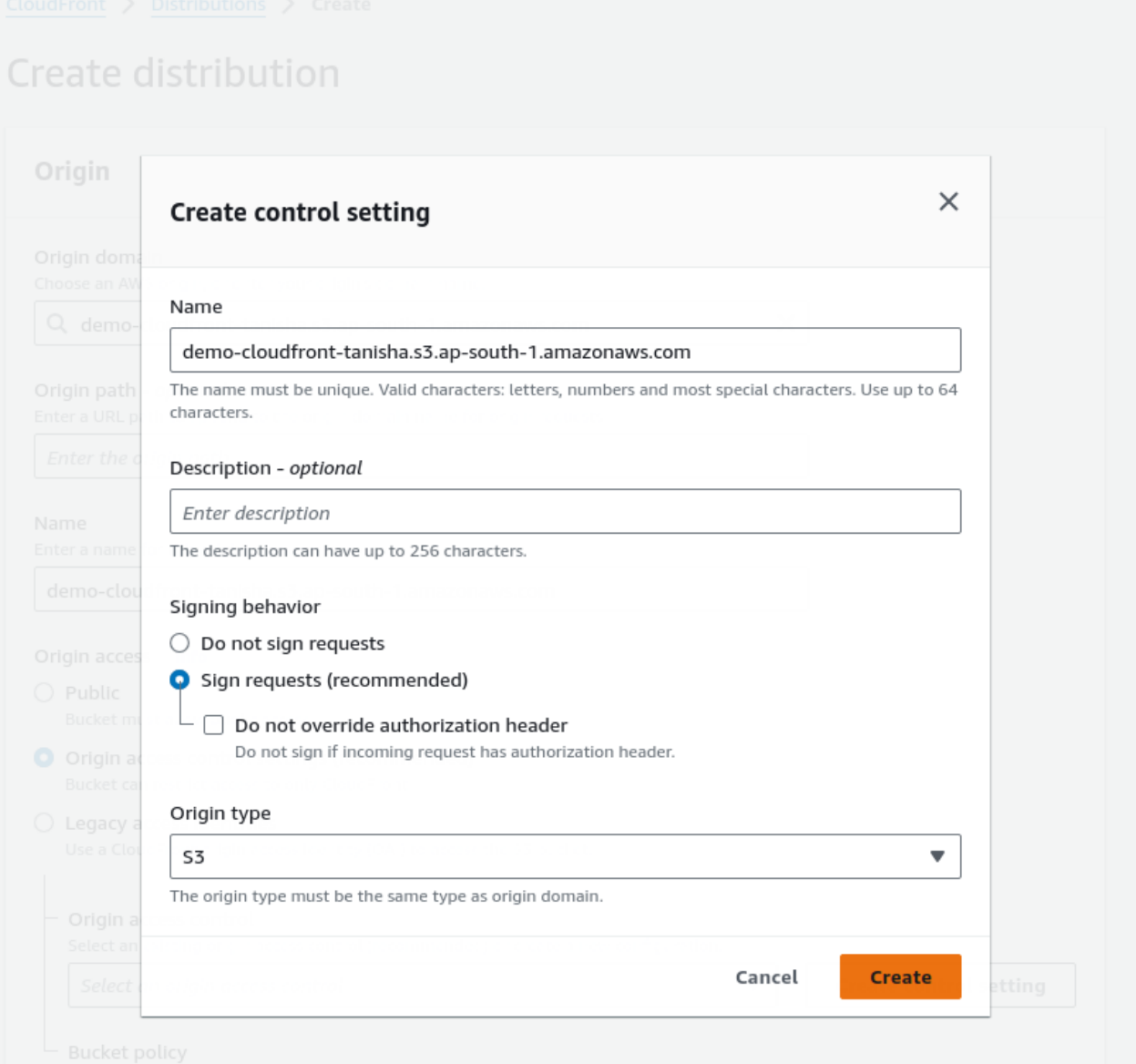
1. Create a S3 bucket to hold our content for our application. Once created, upload some files onto it.
2. Try accessing the index.html file. With object URL, you will get access denied and with the option to open, it will open but the coffee image won't be visible because that object image is not public.

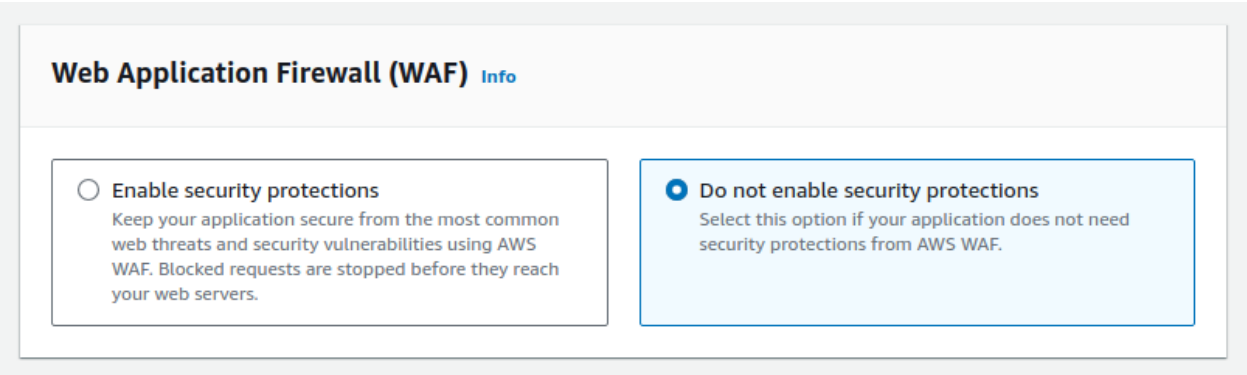

Amazon S3 > Buckets > demo-cloudfront-tanisha > index.html
index.html Info
Copy S3 URI
Download
Open
Object actions
Properties
Permissions
Versions
Object overview
Owner
31fce231227167417e224165b89e400283ebd968120f280f567b2ac269327d97
AWS Region
Asia Pacific (Mumbai) ap-south-1
Last modified
December 8, 2023, 15:32:27 (UTC+05:30)
Size
214.0 B
Type
html
Key
index.html
S3 URI
s3://demo-cloudfront-tanisha/index.html
Amazon Resource Name (ARN)
arn:aws:s3:::demo-cloudfront-tanisha/index.html
Entity tag (Etag)
d59815361dfba53bdaf4668727316864
Object URL
https://demo-cloudfront-tanisha.s3.ap-south-1.amazonaws.com/index.html



3. Lets see how we can use CloudFormation to make these files accessible without making them public.
4. Go to cloudfront, create a distribution.


CloudFront > Distributions > Create
Create distribution
Origin
Origin domain
Choose an AWS origin, or enter your origin's domain name.
demo-cloudfront-tanisha.s3.ap-south-1.amazonaws.com
Origin path - optional Info
Enter a URL path to append to the origin domain name for origin requests.
Enter the origin path
Name
Enter a name for this origin.
demo-cloudfront-tanisha.s3.ap-south-1.amazonaws.com
Origin access Info
Public
Bucket must allow public access.
Origin access control settings (recommended)
Bucket can restrict access to only CloudFront.
Legacy access identities
Use a CloudFront origin access identity (OAI) to access the S3 bucket.
Origin access control
Select an existing origin access control (recommended) or create a new configuration.
Select an origin access control
Create control setting

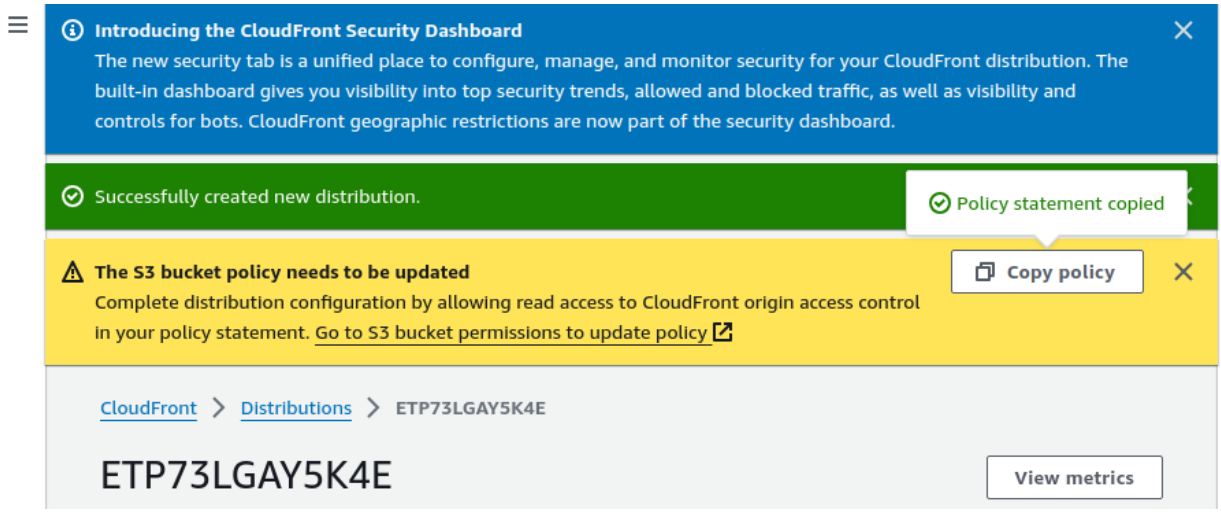

Create distribution
Origin
Create control setting
Name
demo-cloudfront-tanisha.s3.ap-south-1.amazonaws.com
The name must be unique. Valid characters: letters, numbers and most special characters. Use up to 64 characters.
Description - optional
Enter description
The description can have up to 256 characters.
Signing behavior
Do not sign requests
Sign requests (recommended)
Do not override authorization header
Do not sign if incoming request has authorization header.
Origin type
S3
The origin type must be the same type as origin domain.
Cancel
Create

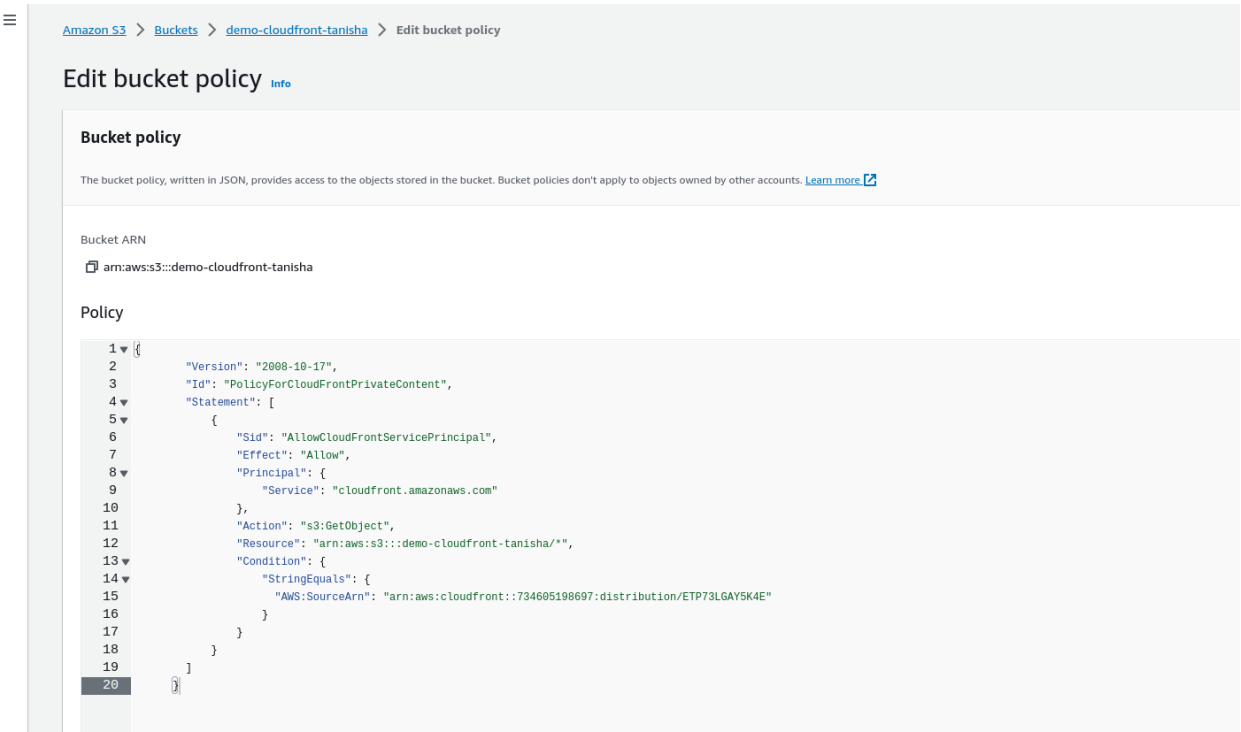

Web Application Firewall (WAF) Info
Enable security protections
Keep your application secure from the most common web threats and security vulnerabilities using AWS WAF. Blocked requests are stopped before they reach your web servers.
Do not enable security protections
Select this option if your application does not need security protections from AWS WAF.

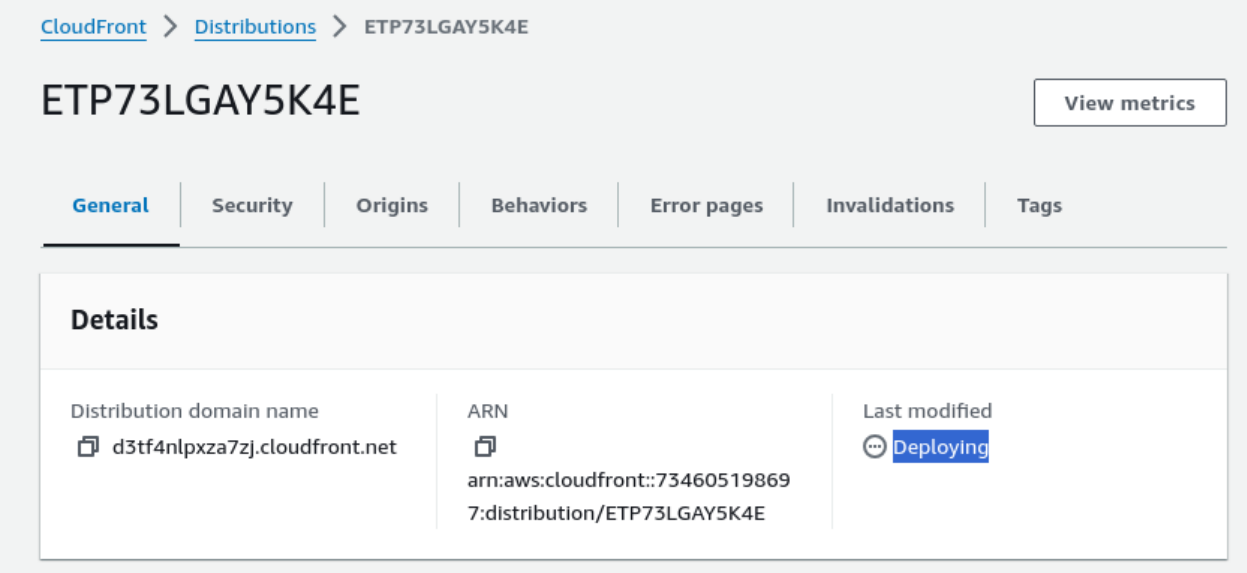
Default root object - *optional*

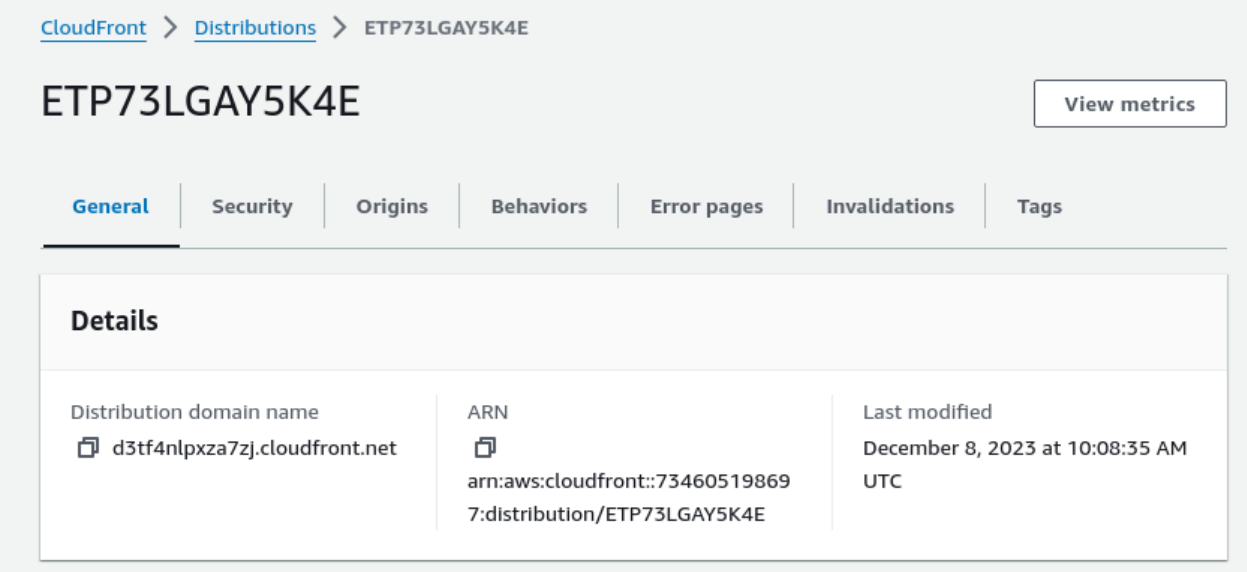
The object (file name) to return when a viewer requests the root URL (/) instead of a specific object.

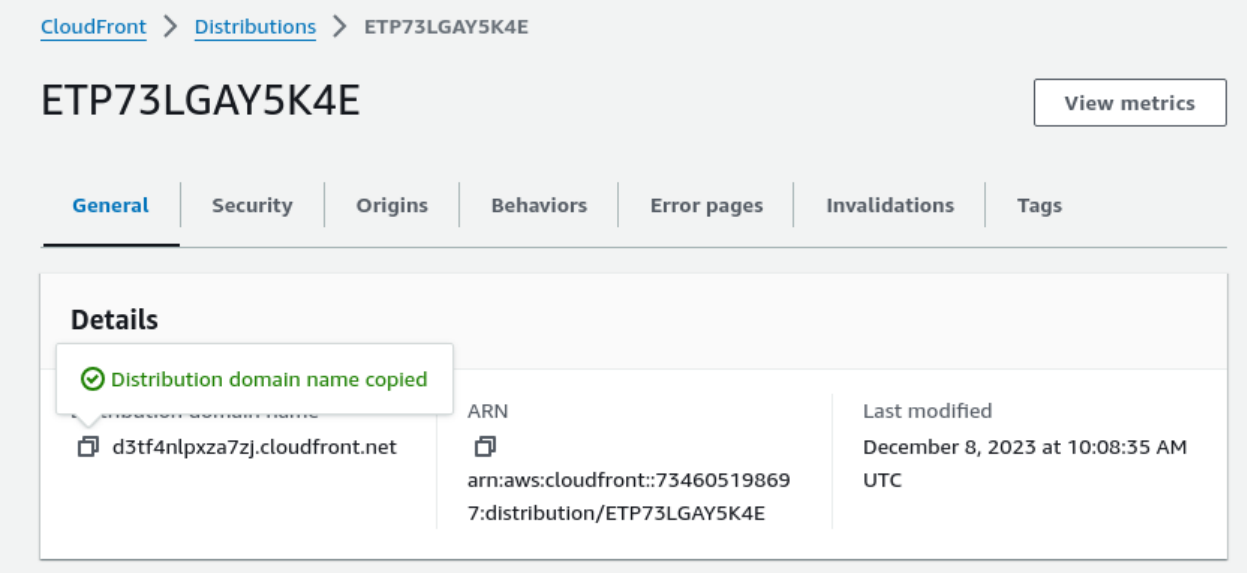
index.html

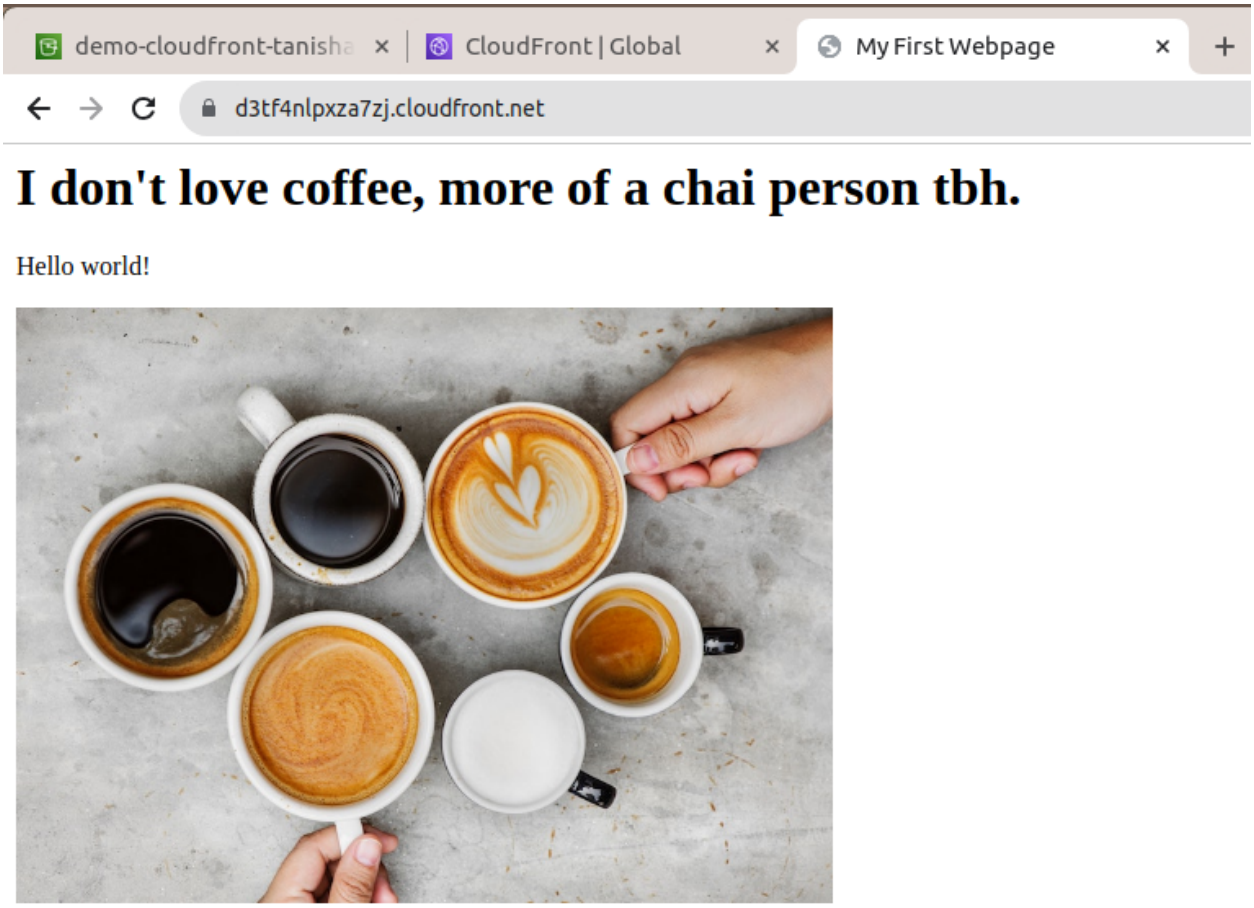





CloudFront > Distributions > ETP73LGAY5K4E
ETP73LGAY5K4E
View metrics
General
Security
Origins
Behaviors
Error pages
Invalidations
Tags
Details
Distribution domain name
d3tf4nlpxza7zj.cloudfront.net
ARN
arn:aws:cloudfront::734605198697:distribution/ETP73LGAY5K4E
Last modified
Deploying


CloudFront > Distributions > ETP73LGAY5K4E
ETP73LGAY5K4E
View metrics
General
Security
Origins
Behaviors
Error pages
Invalidations
Tags
Details
Distribution domain name
d3tf4nlpxza7zj.cloudfront.net
ARN
arn:aws:cloudfront::734605198697:distribution/ETP73LGAY5K4E
Last modified
December 8, 2023 at 10:08:35 AM UTC



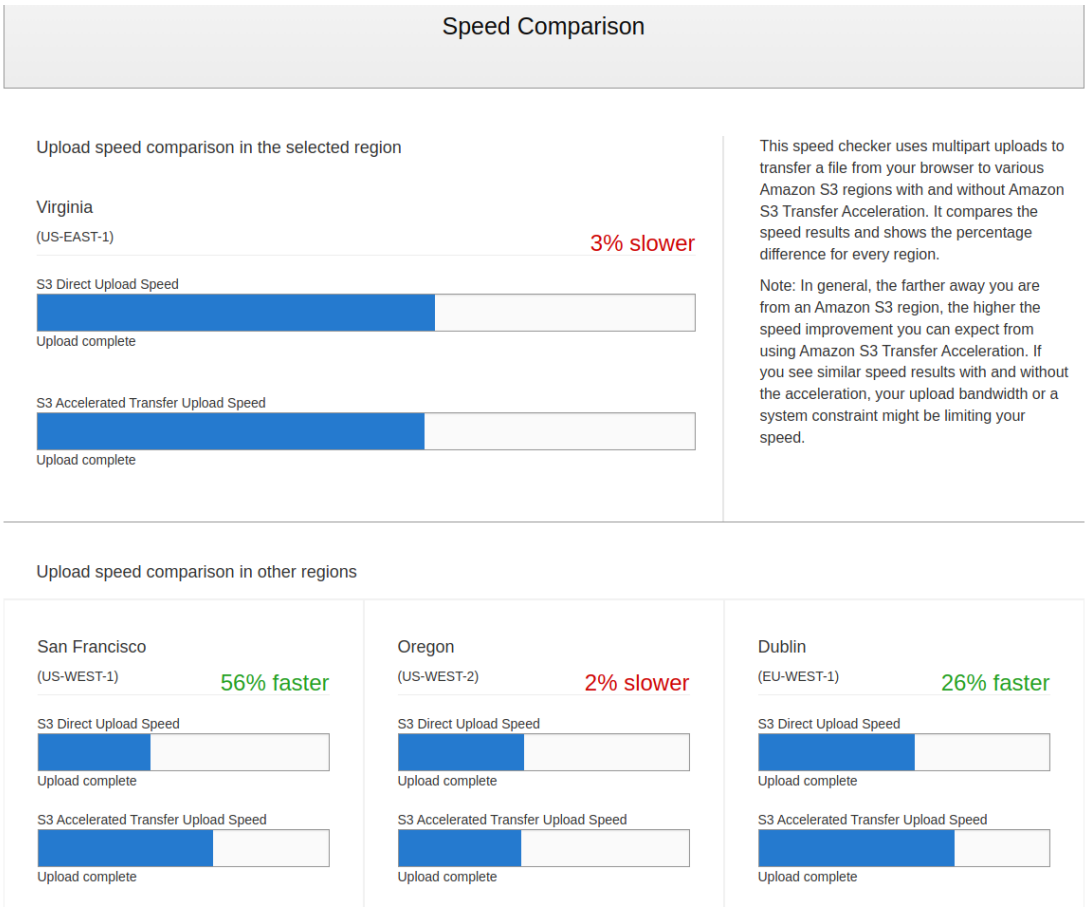


Note: now that you have accessed this webpage, the edge location nearest to you must have cached it. If you refresh now, you will notice that it loads much faster.

S3 Transfer Acceleration

- As we know, a S3 bucket is pinned down to one region and sometimes we need to transfer in data from around the world.
- So, S3 transfer acceleration speeds up this transfer rate by transferring file to an AWS edge location which will forward the data to the S3 bucket in the target region.
- Basically, instead of transferring files directly from region A to B, the user in region A can transfer it to its nearest edge location (fast) and from there, the data can be transferred to the bucket in region B through the private and secure AWS network (also fast). This is overall way faster than transferring from A to B through the internet.

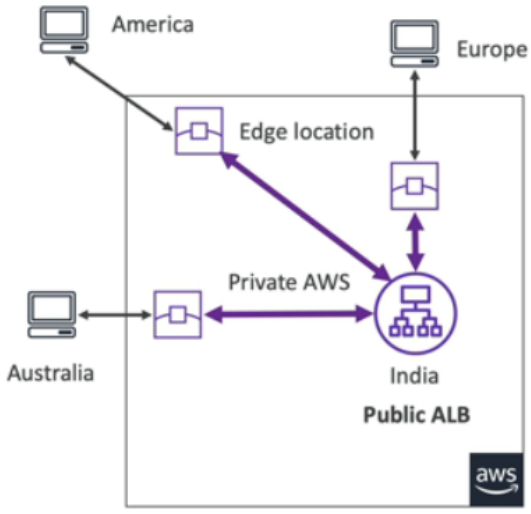
- https://s3-accelerate-speedtest.s3-accelerate.amazonaws.com/en/accelerate-speed-comparsion.html you can check out this website which draws the comparision for you accross multiple regions. It's pretty cool.



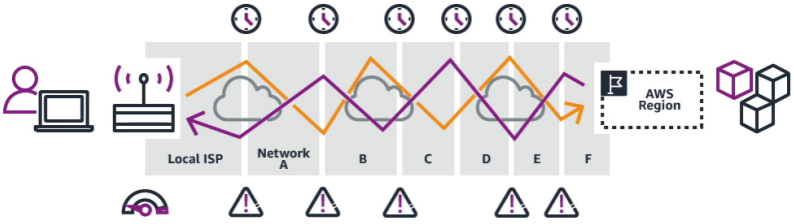
AWS Global Accelerator

AWS Global Accelerator

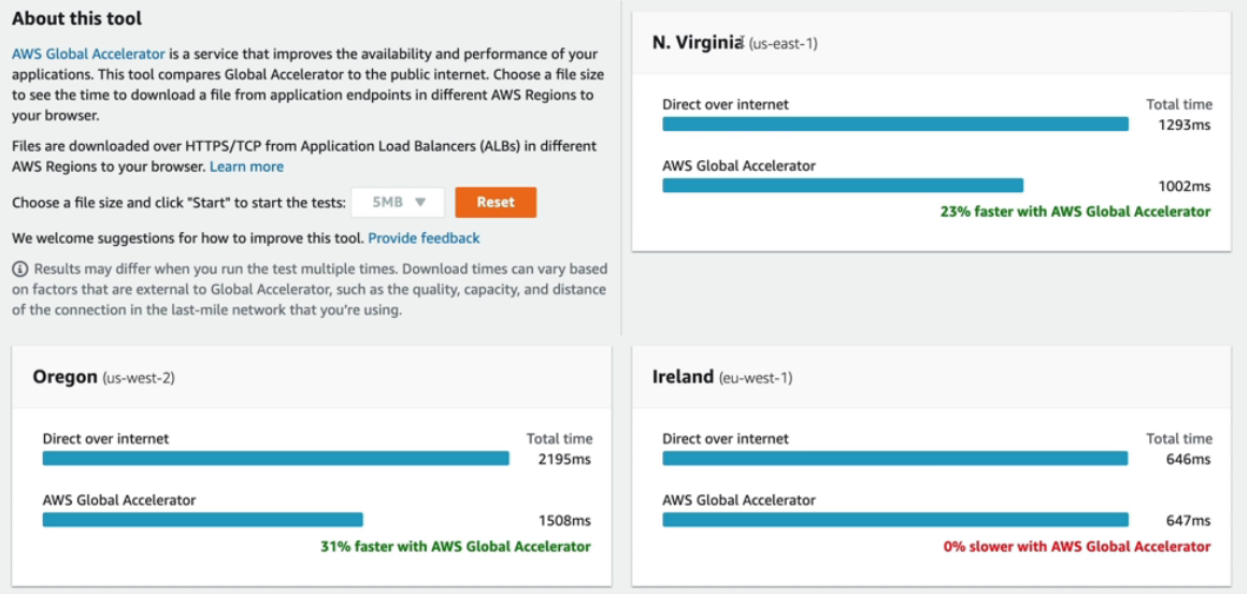
- Improve global application availability and performance using the AWS global network
- Leverage the AWS internal network to optimize the route to your application (60% improvement)
- 2 Anycast IP are created for your application and traffic is sent through Edge Locations
- The Edge locations send the traffic to your application



Without Global Accelerator



With Global Accelerator



AWS Global Accelerator vs CloudFront

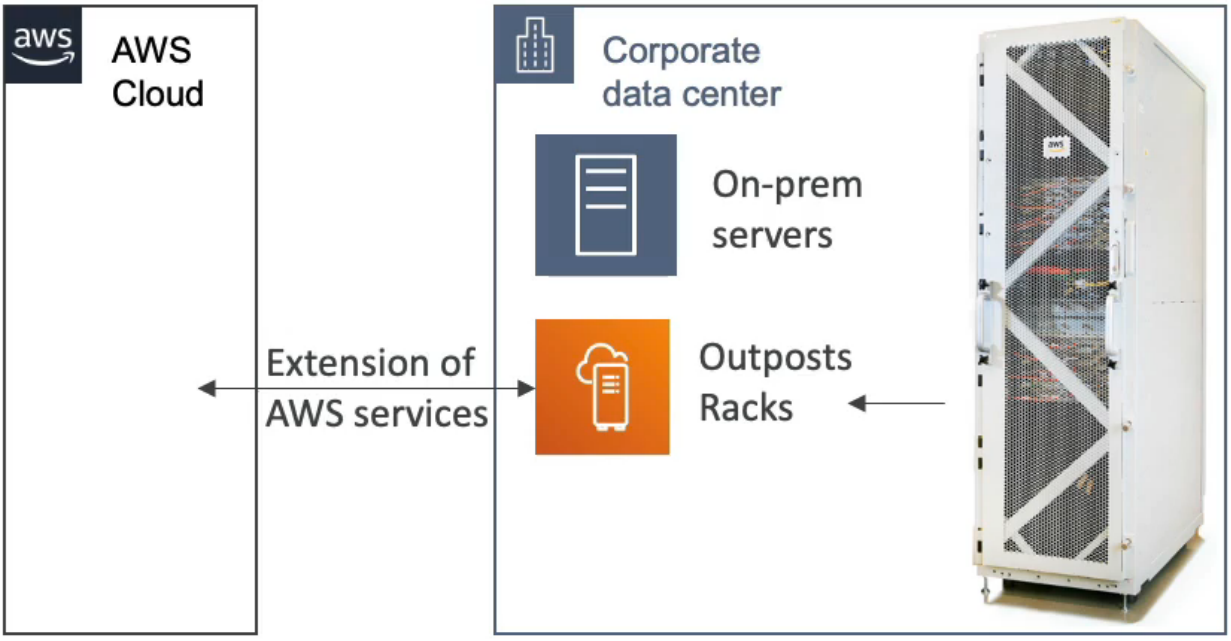
- They both use the AWS global network and its edge locations around the world
- Both services integrate with AWS Shield for DDoS protection.
- CloudFront – Content Delivery Network
  - Improves performance for your cacheable content (such as images and videos)
  - Content is served at the edge
- Global Accelerator
  - No caching, proxying packets at the edge to applications running in one or more AWS Regions.
  - Improves performance for a wide range of applications over TCP or UDP

AWS Outposts

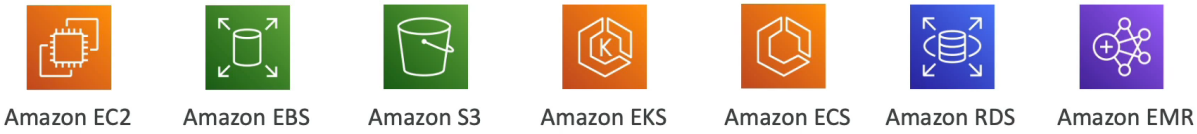
- Some businesses prefer the **hybrid cloud model** which lets them keep their on-premises infrastructure alongside a cloud infrastructure.
- So, there are 2 ways of dealing with this system overall - first is for AWS (through the console, CLI or APIs), second is for their on-premises

infrastructure.
Now, this is 2 different skill sets and might even need 2 different teams to work on, which can be complex.

- Therefore, aws launched Outposts.
- **AWS Outposts is a service that extends AWS infrastructure, services, and tools to on-premises data centers or co-location spaces.**
- Basically, they will come and setup and manage your **Outpost server racks** within your on-premises infrastructure and these servers come preloaded with AWS services and features so you can start leveraging those on-premises.

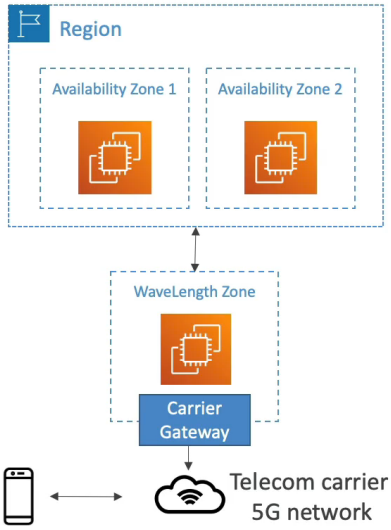


- Benefits:
  - Low-latency access to on-premises systems
  - Local data processing
  - Data residency
  - Easier migration from on-premises to the cloud
  - Fully managed service
- Some services that work on Outposts:



AWS Wavelength

- WaveLength Zones are infrastructure deployments embedded within the telecommunications providers' datacenters at the edge of the 5G networks
- Brings AWS services to the edge of the 5G networks
- Example: EC2, EBS, VPC…
- Ultra-low latency applications through 5G networks
- Traffic doesn't leave the Communication Service Provider's (CSP) network
- High-bandwidth and secure connection to the parent AWS Region
- No additional charges or service agreements
- Use cases: Smart Cities, ML-assisted diagnostics, Connected Vehicles, Interactive Live Video Streams, AR/VR, Real-time Gaming, …

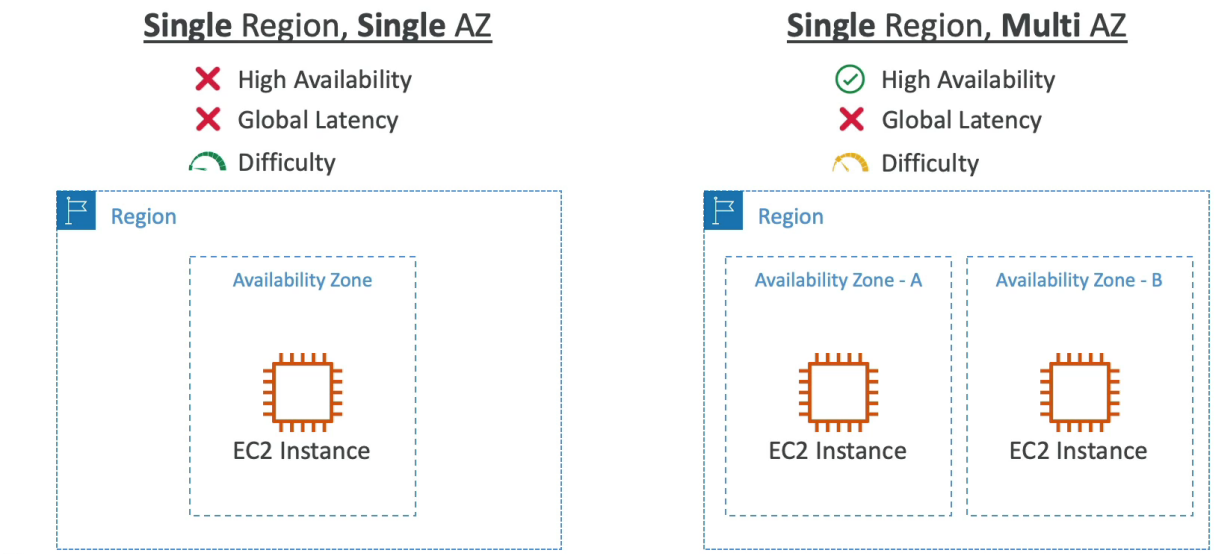


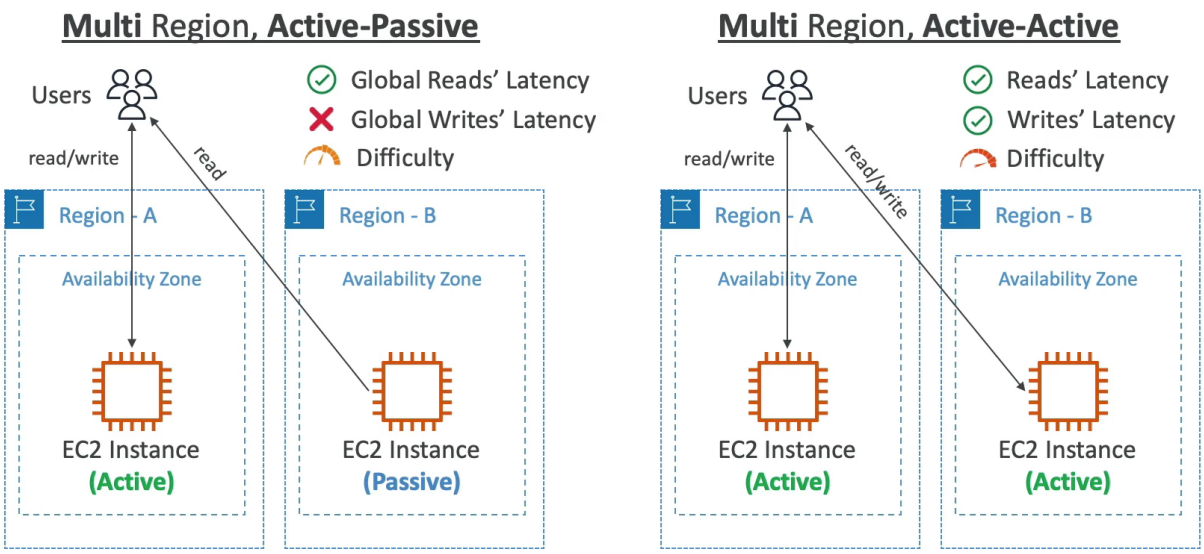
AWS Local Zones

- AWS Local Zones are like mini-extensions of AWS regions, but they are located closer to specific cities. Think of them as small data center outposts.
- You can basically place your AWS compute, storage, database and other such services close to the end users to run latency-sensitive applications.
- Regular AWS regions are in specific geographic areas, but Local Zones are set up near major cities. This brings AWS services physically closer to where people and businesses are located.
- An extension of an AWS Region.

- Example:
  - AWS Region: N.Virginia (us-east-1)
  - AWS Local Zones: Boston, Chicago, Dallas, Houston, Miami, ...

Architecture for Global Applications





Summary

- Global DNS: Route 53
  - Great to route users to the closest deployment with least latency
  - Great for disaster recovery strategies
- Global Content Delivery Network (CDN): CloudFront
  - Replicate part of your application to AWS Edge Locations – decrease latency
  - Cache common requests – improved user experience and decreased latency
- S3 Transfer Acceleration
  - Accelerate global uploads & downloads into Amazon S3
- AWS Global Accelerator
  - Improve global application availability and performance using the AWS global network

- AWS Outposts
  - Deploy Outposts Racks in your own Data Centers to extend AWS services

- AWS WaveLength
  - Brings AWS services to the edge of the 5G networks
  - Ultra-low latency applications

- AWS Local Zones
  - Bring AWS resources (compute, database, storage, …) closer to your users
  - Good for latency-sensitive applications

Good job!
Weighted Routing Policy is used to route traffic to multiple resources in proportions that you specify.

Question 1:

Which Route 53 Routing Policies would you use to route traffic to multiple resources in proportions that you specify?

- ○ Simple Routing Policy
- ◉ **Weighted Routing Policy**
- ○ Latency Routing Policy
- ○ Failover Routing Policy

Good job!
CloudFront uses Edge Location to cache content, and therefore bring more of your content closer to your viewers to improve read performance.

Question 4:

What does AWS CloudFront use to improve read performance?

- ○ DDoS Protection
- ○ S3 Buckets Fast-Read
- ◉ **Caching Content in Edge Locations**
- ○ Caching Content in Edge Regions

Good job!
A global application is not specifically used to scale elastically on demand. You can use Auto Scaling Groups for example if you want to elastically scale based on demand.

Question 6:

Which of the following statements is NOT a reason for a global application?

- ○ Decreased Latency
- ○ Disaster Recovery
- ◉ **Scale elastically on demand**
- ○ Attack protection

Good job!
Route 53 features are (non exhaustive list): Domain Registration, DNS, Health Checks, Routing Policy

Question 7:

Which features are available with Route 53?

- ○ Health Checks, Auto Scaling, Routing Policy, DNS
- ○ Load Balancing, DNS, Domain Registration, Monitoring
- ○ Domain Registration, DNS, Health Checks, DDoS Protection
- ◉ **Domain Registration, DNS, Health Checks, Routing Policy**

Good job!
You can use AWS WAF web access control lists (web ACLs) to help minimize the effects of a distributed denial of service (DDoS) attack. For additional protection against DDoS attacks, AWS also provides AWS Shield Standard and AWS Shield Advanced.

Question 8:

With which services does CloudFront integrate to protect against web attacks?

- ◉ **WAF & Shield**
- ○ WAF & IAM
- ○ IAM & Shield
- ○ Security Groups & WAF