Global infrstructure

**Why make a global application?**

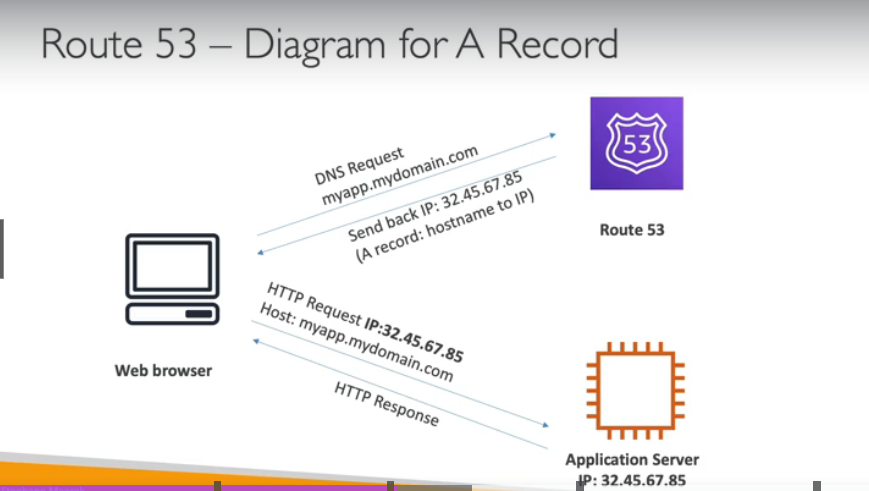
* A global application is an application deployed in **multiple geographies**
* On AWS: this could be **Regions** and / or **Edge Locations**
* **Decreased Latency**
  + Latency is the time it takes for a network packet to reach a server
  + It takes time for a packet from Asia to reach the US
  + Deploy your applications closer to your users to decrease latency, better experience
* **Disaster Recovery (DR)**
  + If an AWS region goes down (earthquake, storms, power shutdown, politics)…
  + You can fail-over to another region and have your application still working
  + A DR plan is important to increase the availability of your application
* **Attack protection**: distributed global infrastructure is harder to attack

### Global AWS Infrastructure

* Regions: For deploying applications and infrastructure
* Availability Zones: Made of multiple data centers
* Edge Locations (Points of Presence): for content delivery as close as possible to users
* More at: <https://infrastructure.aws/>

**Amazon Route 53 Overview**

* Route53 is a Managed DNS (Domain Name System)
* Route 53 is essentially an address book for the internet. It's a service offered by Amazon Web Services (AWS) that translates user-friendly domain names (like <https://www.example.com>) into the numerical IP addresses computers use to connect to websites..
* In AWS, the most common records are:
  + [www.google.com](http://www.google.com/) => 12.34.56.78 == A record (IPv4)
  + [www.google.com](http://www.google.com/) => 2001:0db8:85a3:0000:0000:8a2e:0370:7334 == AAAA IPv6



## Route 53 Routing Policies

**Simple Routing:**

* Best for single resources.
* Route 53 returns all available IP addresses to the user's device, which picks one at random (no health checks).

**Weighted Routing:**

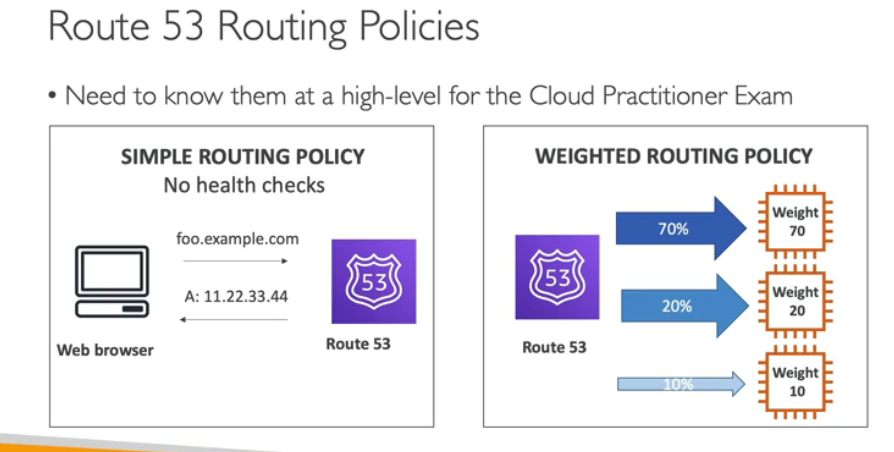
* Distributes traffic among resources with assigned weights.
* Higher weight means more traffic directed to that resource (useful for load balancing).

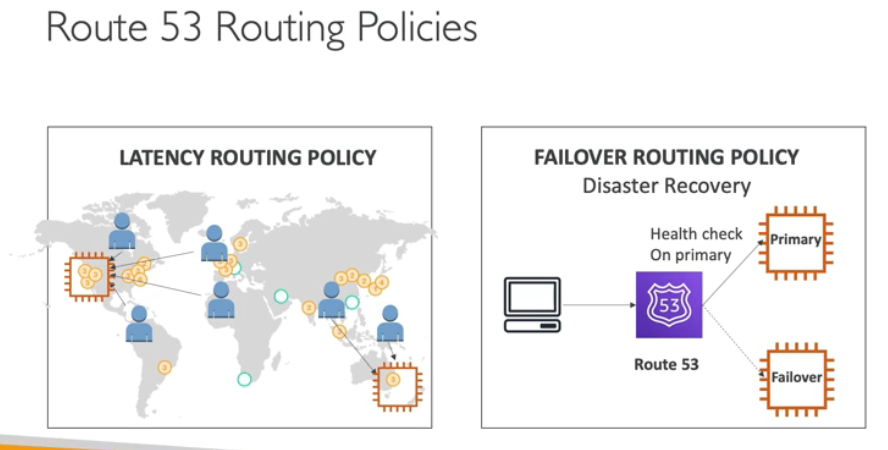
**Latency Routing:**

* Routes users to the resource with the lowest latency (response time) from their location.
* Ideal for geographically distributed websites where speed matters.

**Failover Routing:**

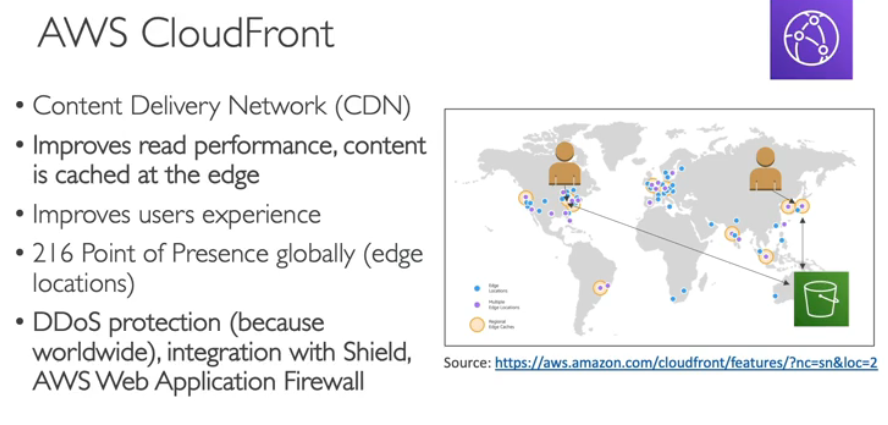
* Sets up a primary and a backup resource.
* If the primary fails a health check, Route 53 directs traffic to the backup.
* Ensures high availability for critical applications.





**AWS CloudFront**

* Content Delivery Network (CDN)
* **Improves read performance, content is cached at the edge**
* Improves users experience
* 216 Point of Presence globally (edge locations)
* DDoS protection (because worldwide), integration with Shield, AWS Web Application Firewall
* Source: <https://aws.amazon.com/cloudfront/features/?nc=sn&loc=2>



Imagine you have a store that sells cookies (your website). People from all over the world come to your store to buy cookies (users visiting your website). Normally, you would have to bake all the cookies yourself and give them to each person (your origin server has to deliver all the content).

CloudFront is like having a bunch of bakeries around the world (edge locations). People who live close to one of your bakeries can buy cookies from there instead of having to travel all the way to your store (users can get content from the nearest edge location instead of the origin server). This way, people get their cookies faster (users experience faster loading times).

### CloudFront - Origins

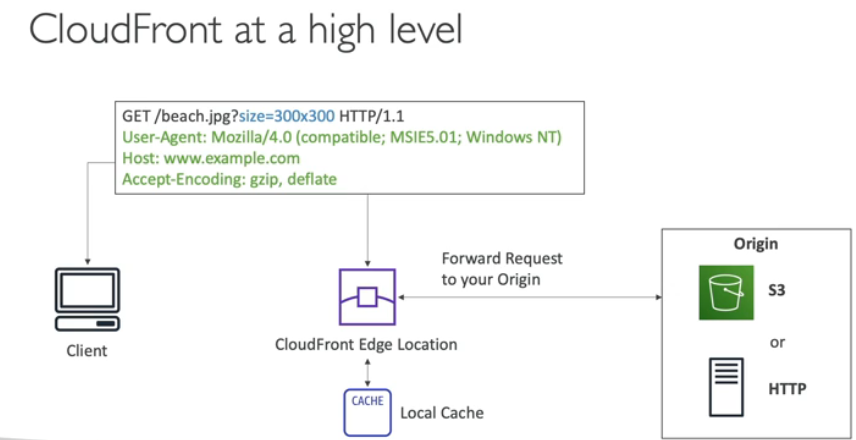
 **S3 bucket:** This is a common choice for storing static content like images, videos, or website assets. CloudFront can cache this content at its edge locations, significantly improving loading times for users.

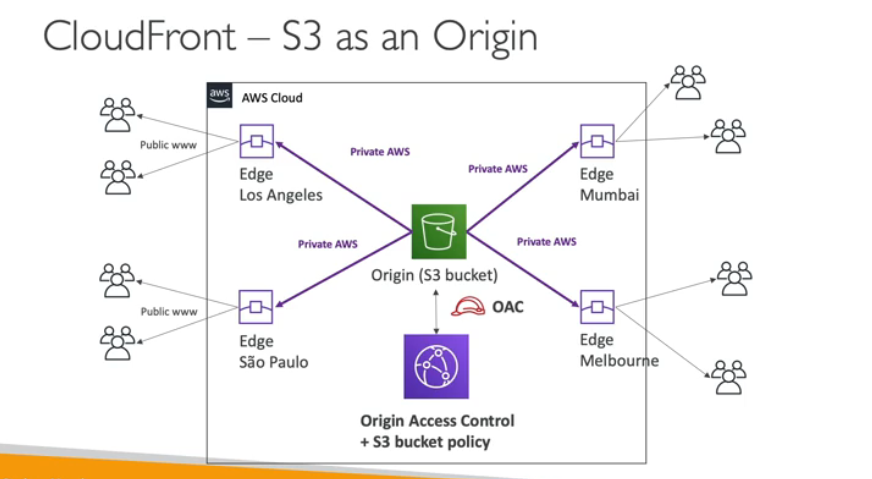
 **S3 bucket with CloudFront Origin Access Identity (OAI):** This adds a security layer by restricting access to your S3 bucket. Users can only access the content through CloudFront's signed URLs, not by directly accessing the bucket.

 **Custom Origin (HTTP):** This allows you to use any HTTP server as an origin, including:

* **Application Load Balancer:** Distributes traffic among multiple web servers for high availability and scalability.
* **EC2 instance:** An Amazon virtual server where you can run your own web application.
* **Any HTTP backend:** This gives you flexibility to use any web server as long as it's accessible via HTTP.

 **S3 website (static website hosting):** If your S3 bucket is configured as a static website, you can use it directly as a CloudFront origin.



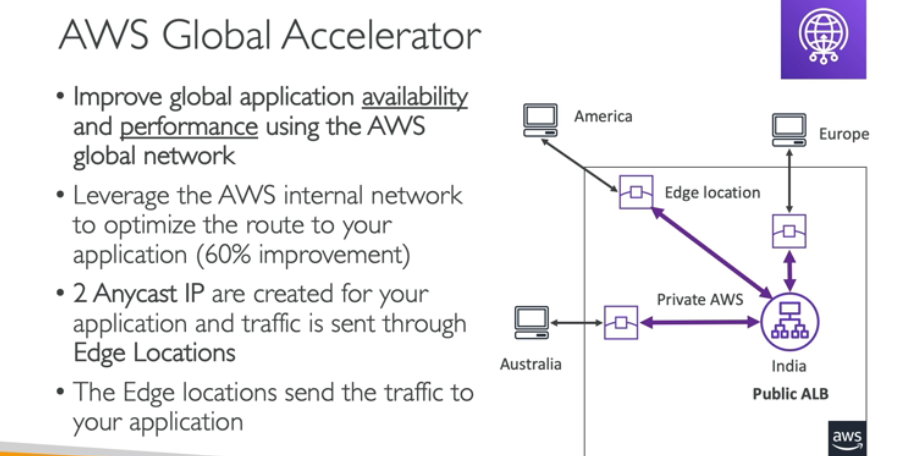


### CloudFront vs S3 Cross Region Replication

| **CloudFront** | **S3 Cross Region Replication** |
| --- | --- |
| Global Edge network | Must be setup for each region you want replication to happen |
| Files are cached for a TTL (Time to Live) (maybe a day) | Files are updated in near real-time, Read only |
| **Great for static content that must be available everywhere** | **Great for dynamic content that needs to be available at low-latency in few regions** |

### S3 Transfer Acceleration

* Increase transfer speed by transferring file to an AWS edge location which will forward the data to the S3 bucket in the target region
* if we try to upload file to Australia S3 bucket it will take time using CloudFront we can rescue time.
* File in USA -> Edge Location(USA) -> S3 Bucket(Australia)



### 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
  + Good for HTTP use cases that require static IP addresses
  + Good for HTTP use cases that required deterministic, fast regional failover

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23