CloudFront & Global Accelerator

Amazon CloudFront

- Content Delivery Network (CDN)
- **Improves read performance**, content is cached at the edge
- Improves user experience
- 216 Points of Presence globally (edge locations)
- DDoS protection (because of global distribution), integration with AWS Shield and AWS Web Application Firewall

How it works (visual explanation)

CloudFront uses a globally distributed network of **edge locations** to cache and deliver content closer to the users. The map shows:

- **Edge Locations** (blue dots): servers closer to the users for delivering content.
- **Multiple Edge Locations** (purple circles): regions with high density of edge servers.
- **Regional Edge Caches** (orange rings): larger caches used to serve content not yet stored at the edge.

Users across different continents connect to the nearest edge location instead of the origin server (e.g., an S3 bucket), reducing latency and improving performance.

Source: AWS CloudFront Features

CloudFront – Origins

S3 Bucket

- For distributing files and caching them at the edge
- For uploading files to S3 through CloudFront
- Secured using Origin Access Control (OAC)

VPC Origin

- For applications hosted in VPC private subnets
- Application Load Balancer / Network Load Balancer / EC2 Instances

Custom Origin (HTTP)

- S3 website (must first enable the bucket as a static S3 website)
- Any public HTTP backend you want

CloudFront at a High Level

How a request flows

1. A **client** makes an HTTP request:

GET /beach.jpg?size=300x300 HTTP/1.1 User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT) Host: www.example.com Accept-Encoding: gzip, deflate

2. The request is routed to a **CloudFront Edge Location**.

3. If the content is in the **local cache**, it's returned immediately to the client.

4. If not, CloudFront **forwards the request to the origin**, which can be:

- **S3** bucket
- **HTTP server** (e.g., EC2, Load Balancer, on-premises server)

5. The response from the origin is cached at the edge for future requests, improving performance and reducing load on the origin.

This architecture minimizes latency and accelerates content delivery by serving requests from edge locations closest to users.

CloudFront – S3 as an Origin

How it works

- Users from around the world access content through **public CloudFront URLs**.
- Requests are routed to the nearest **CloudFront Edge Location** (e.g., Los Angeles, São Paulo, Mumbai, Melbourne).
- The edge locations communicate **privately within AWS** with the origin, which is an **S3 bucket**.

Security

- Access to the S3 bucket is restricted using:
  - **Origin Access Control (OAC)**
  - **S3 bucket policy**

This ensures that the S3 bucket is **not publicly accessible**, and only CloudFront is allowed to fetch the content through secure, private AWS connections.

CloudFront vs S3 Cross Region Replication

CloudFront

- Global Edge network
- Files are cached for a TTL (maybe a day)
- **Great for static content that must be available everywhere**

S3 Cross Region Replication

- Must be set up for each region you want replication to happen
- Files are updated in near real-time
- Read only
- **Great for dynamic content that needs to be available at low-latency in few regions**

CloudFront – ALB or EC2 as an Origin (Using VPC Origins)

Key Concepts

- Allows you to deliver content from applications hosted in **VPC private subnets**.
- No need to expose these resources directly to the internet.

Deliver traffic to private:

- **Application Load Balancer (ALB)**
- **Network Load Balancer (NLB)**
- **EC2 Instances**

Architecture Summary

1. **Users** send requests to CloudFront.
2. **CloudFront Edge Location** routes the request to the **VPC Origin**.
3. Inside the VPC, traffic is forwarded to the corresponding **private resources** (ALB, NLB, or EC2).
4. This setup improves **security** by keeping services private and enhances performance via edge caching.

CloudFront – ALB or EC2 as an Origin Using Public Network

Key Concepts

When using **public network access** (instead of VPC origins):

- You must **allow the public IPs of CloudFront Edge Locations** in your origin's security groups.
- Reference for CloudFront IPs:
  CloudFront IP List

EC2 Instance as Origin

- **EC2 Instances must be Public**
- Must allow traffic from **CloudFront IPs** in the security group.

ALB as Origin

- **Application Load Balancer must be Public**
- Must allow traffic from **CloudFront IPs**
- **EC2 Instances behind the ALB can be private**
- EC2 security group must allow traffic from the **ALB's security group**

Summary

Using the public network allows CloudFront to access your resources directly over the Internet, but it requires careful configuration of public IP access and security group rules to avoid exposure.

CloudFront Geo Restriction

Restrict Access Based on Country

- You can control who can access your distribution using country-based restrictions.

Options:

- **Allowlist**:
  Only allow users located in countries that are explicitly approved.

- **Blocklist**:
  Deny access to users from countries on a list of banned locations.

Additional Notes:

- The "country" is determined using a **third-party Geo-IP database**.
- **Use Case**: Often used to enforce **copyright laws** by controlling content distribution based on geography.

CloudFront – Pricing

- CloudFront Edge locations are distributed globally.
- **Data transfer costs vary by region and usage volume.**

Example Pricing (Per GB, based on monthly usage)

Usage Tier	US, Canada	Europe	Africa & ME	S. America	Japan	Aus/NZ	SE Asia	India
First 10TB	$0.085	$0.085	$0.110	$0.110	$0.114	$0.114	$0.140	$0.170
Next 40TB	$0.080	$0.080	$0.105	$0.105	$0.089	$0.098	$0.135	$0.130
Next 100TB	$0.060	$0.060	$0.090	$0.090	$0.086	$0.094	$0.120	$0.110
Next 350TB	$0.040	$0.040	$0.080	$0.080	$0.084	$0.092	$0.100	$0.100
Next 524TB	$0.030	$0.030	$0.060	$0.060	$0.080	$0.090	$0.080	$0.100
Next 4PB	$0.025	$0.025	$0.050	$0.050	$0.070	$0.085	$0.070	$0.100
Over 5PB	$0.020	$0.020	$0.040	$0.040	$0.060	$0.080	$0.060	$0.100

> *Costs **decrease** with higher usage and vary **significantly** between regions.*

CloudFront – Price Classes

Cost Optimization Through Edge Location Selection

You can **reduce the number of edge locations** used by your distribution to **lower costs**.

Three Price Classes

1. **Price Class All**:
    - Includes **all regions**
    - **Best performance**, highest cost
2. **Price Class 200**:
    - Includes **most regions**, **excludes the most expensive**
    - Balanced performance and cost
3. **Price Class 100**:
    - Includes only the **least expensive regions**
    - Best for **cost savings**, lower performance

Region Coverage by Price Class

Region	Price Class All	Price Class 200	Price Class 100
United States, Mexico, & Canada	Yes	Yes	Yes
Europe & Israel	Yes	Yes	Yes

South Africa, Kenya, & Middle East	Yes	Yes	No
South America	Yes	No	No
Japan	Yes	Yes	No
Australia & New Zealand	Yes	No	No
Hong Kong, Philippines, Singapore, etc.	Yes	Yes	No
India	Yes	Yes	No

CloudFront – Cache Invalidations

Problem

- When the **origin (e.g., S3 bucket)** is updated, CloudFront **does not immediately fetch the new content**.
- CloudFront will continue to serve cached content until the **TTL (Time to Live)** expires.

Solution

- You can **force a cache refresh** before the TTL expires by performing a **CloudFront Invalidation**.

How it works

- You can invalidate:
  - All files using `/*`
  - A specific path such as `/index.html`
  - A directory pattern like `/images/*`

Use Case Example

- You update `index.html` and files in `/images/` in your S3 bucket.
- You submit an invalidation request for:
  - `/index.html`
  - `/images/*`
- CloudFront removes the outdated files from all edge caches and fetches updated versions from the origin.

This mechanism ensures users receive **fresh content** even before the cache naturally expires.

Global Users for Our Application

Scenario

- You've deployed an application and have **global users** who want to access it.
- These users connect over the **public internet**, which introduces **latency** due to **multiple hops** across various network segments.

Problem

- The more network hops between the user and your origin (e.g., a **Public ALB** in India), the **slower** the connection.
- Latency increases especially for users located far from the application's origin region (e.g., users in America or Australia accessing a service in India).

Goal

- **Minimize latency** by routing traffic through the **AWS global network** instead of the public internet.
- This is a foundational reason to adopt services like **CloudFront** or **Global Accelerator**, which help **optimize routing** and **reduce the number of hops**.

Unicast IP vs Anycast IP

Unicast IP

- Each **server has a unique IP address**.
- The client connects to a specific IP corresponding to one server.
- Example:
  - Server A → 12.34.56.78
  - Server B → 98.76.54.32
  - The client must choose which specific IP to contact.

Anycast IP

- **Multiple servers share the same IP address**.
- The client is automatically **routed to the nearest (or best) server** using network topology.
- Example:
  - Server A, B, and C all respond to **12.34.56.78**
  - The client is routed to the **closest server** (e.g., lowest latency)

Summary

- **Unicast** = one IP per server
- **Anycast** = one IP shared across many servers, with routing based on proximity

AWS Global Accelerator

Key Features

- **Leverages the AWS internal network** to route traffic to your application, reducing latency and improving availability.
- **Two Anycast IPs** are automatically created for your application.

How it works

1. Clients (e.g., in America, Europe, Australia) connect to **Anycast IPs**.
2. These IPs route the traffic to the **nearest AWS Edge Location**.
3. From there, traffic is sent through the **AWS private network** directly to your application (e.g., Public ALB in India).

Benefits

- Uses **Anycast routing** for optimal performance.
- Minimizes the number of hops and latency by avoiding the public internet.
- Ensures **fast, reliable access** for global users by routing through AWS infrastructure.

AWS Global Accelerator

Supported Resources

- Works with:
  - **Elastic IP**
  - **EC2 instances**

    - **Application Load Balancer (ALB)**
    - **Network Load Balancer (NLB)**
    - Both **public and private** endpoints

Consistent Performance

- Intelligent routing to **lowest latency** and **fast regional failover**
- No caching issues on the client side (IP stays the same)
- Uses the **internal AWS network**

Health Checks

- Global Accelerator performs **health checks** on your endpoints
- Enables **automatic failover** in less than 1 minute if a target is unhealthy
- Useful for **disaster recovery** scenarios

Security

- Only **2 external IPs** need to be whitelisted
- **DDoS protection** is included via **AWS Shield**

AWS Global Accelerator vs CloudFront

Common Features

- Both use the **AWS global network** and its **edge locations** worldwide.
- Both integrate with **AWS Shield** for **DDoS protection**.

CloudFront

- Improves performance for **cacheable content** (e.g., images, videos).
- Supports **dynamic content** (e.g., API acceleration, dynamic site delivery).
- **Content is served at the edge**, closest to the user.

Global Accelerator

- Improves performance for **a wide range of applications** over **TCP or UDP**.
- **Proxies packets** from the edge to backend applications in one or more AWS regions.
- Ideal for **non-HTTP use cases** like:
    - Gaming (UDP)
    - IoT (MQTT)
    - Voice over IP
- Also suited for **HTTP** use cases that:
    - Require **static IP addresses**
    - Require **deterministic, fast regional failover**