aws
certified
Solutions Architect
Professional
DOLFINed

AMAZON ROUTE 53 & CLOUDFRONT

AMAZON ROUTE 53

Amazon Route53

Domain Registration with Route 53

Amazon ROUTE 53

ROUTE53

Route53 performs three main functions:

a) Register a domain

b) As a DNS, it routes Internet traffic to the resources for your domain

c) Check the health of your resources –

- Route53 sends automated requests over the internet to a resource (can be a web server) to verify that the server is, reachable, functional, available.
- Also, you can choose to receive notifications when a resource becomes unavailable and choose to route internet traffic away from unhealthy resources.

- You can use AWS Route53 for any any combination of these functions.

Amazon ROUTE 53

Supported DNS Record Types

Supported DNS Record types by ROUTE 53

- **A Record –** Address Record – Maps domain name to IP address
  - www.dolfined.com IN A 2.2.2.2
- **AAAA Record -** IPv6 Address Record – Maps domain name to an IPv6 address
  - www.dolfined.com IN AAAA 2001:d8b1::1
- **CNAME Record –** Maps an alias to a hostname
  - web IN CNAME www.dolfined.com
- **NS Record –** Name server record – Used for delegating zone to a nameserver
  - dolfined.com IN NS ns1.dolfined.com
- **SOA Record –** Start of Authority Record -
- **MX Record –** Mail Exchanger – Defines where to deliver mail for user @ a domain name
  - dolfined.com IN MX 10 mail01.dolfined.com
    IN MX 20 mail02.dolfined.com
- CAA, PTR, NAPTR, SPF, SRV, TXT Records

CNAME Record

CNAME Record Type

A CNAME Value element is the same format as a domain name.

- The DNS protocol does not allow you to create a CNAME record for the top node of a DNS namespace, also known as the zone apex (or Root Domain, or Naked Domain).
  - For example, if you register the DNS name dolfined.com, the zone apex is dolfined.com,
    - You cannot create a CNAME record for dolfined.com,
  - However, you can create CNAME records for www.dolfined.com, support.dolfined.com, and so on.

- In addition, if you create a CNAME record for a subdomain, you cannot create any other records for that subdomain.
  - For example, if you create a CNAME for www.dolfined.com,
    - You cannot create any other records for which the value of the Name field is www.dolfined.com.

Amazon ROUTE 53

Alias Record

- Specific to Route 53 and not seen outside.

- You can use it to create DNS Route53 Records and route queries to AWS services the IP address of which can change (CLB/ALB/NLB, CloudFront Distribution, S3 Bucket configured as a static website, ElasticBeanStalk environment, API Gatewaty, VPC interface endpoint, Global Accelerator accelerator, and another route53 record in the same hosted zone.
  - When you point an Alias to one of these AWS services, Route53 will fetch the IP address of that service's resource(s) in real time to respond to DNS queries

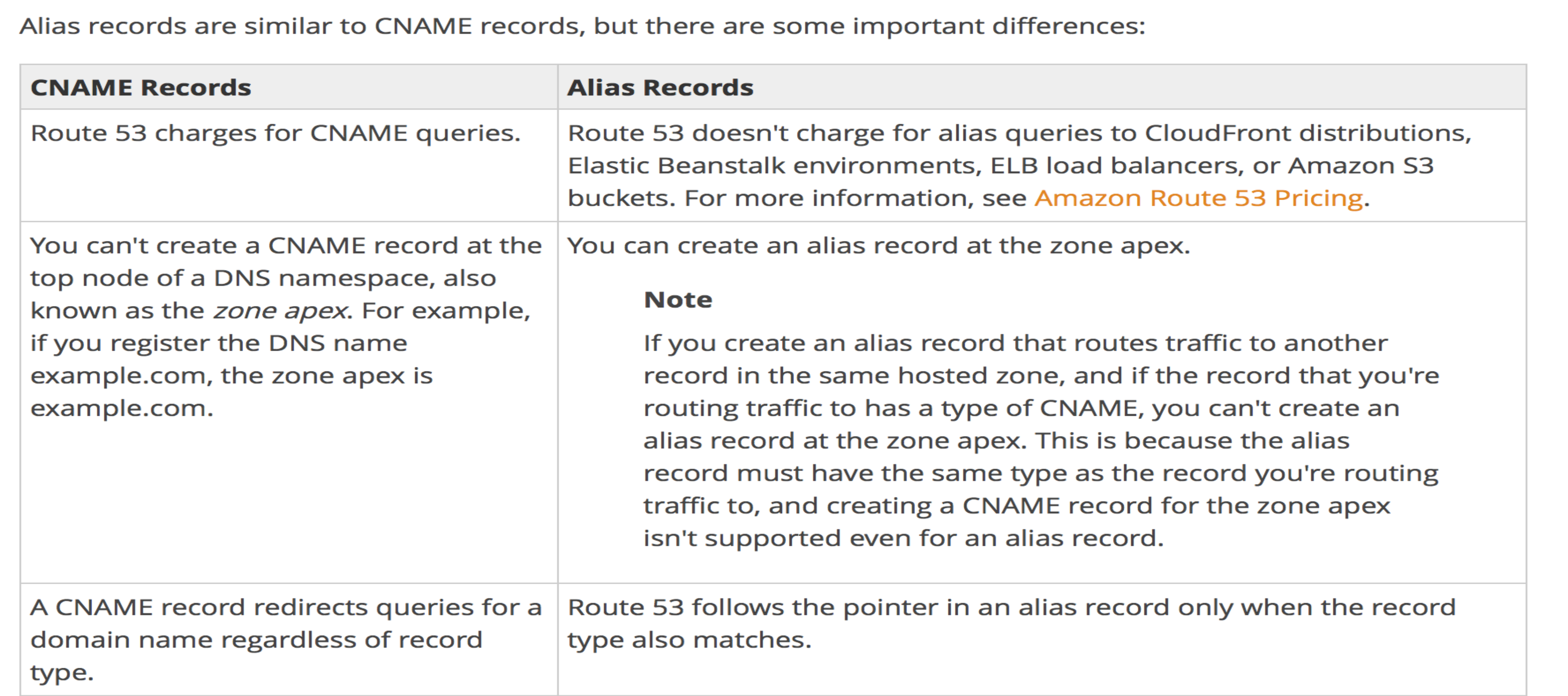
- You CAN'T create a CNAME for the apex/naked/root domain name,
  - Alias Record CAN do that
- You CAN NOT Alias to a record or resource outside of AWS Route 53 or AWS Services

Amazon Route53

ALIAS vs CNAME Records

Amazon ROUTE 53

CNAME vs ALIAS Records

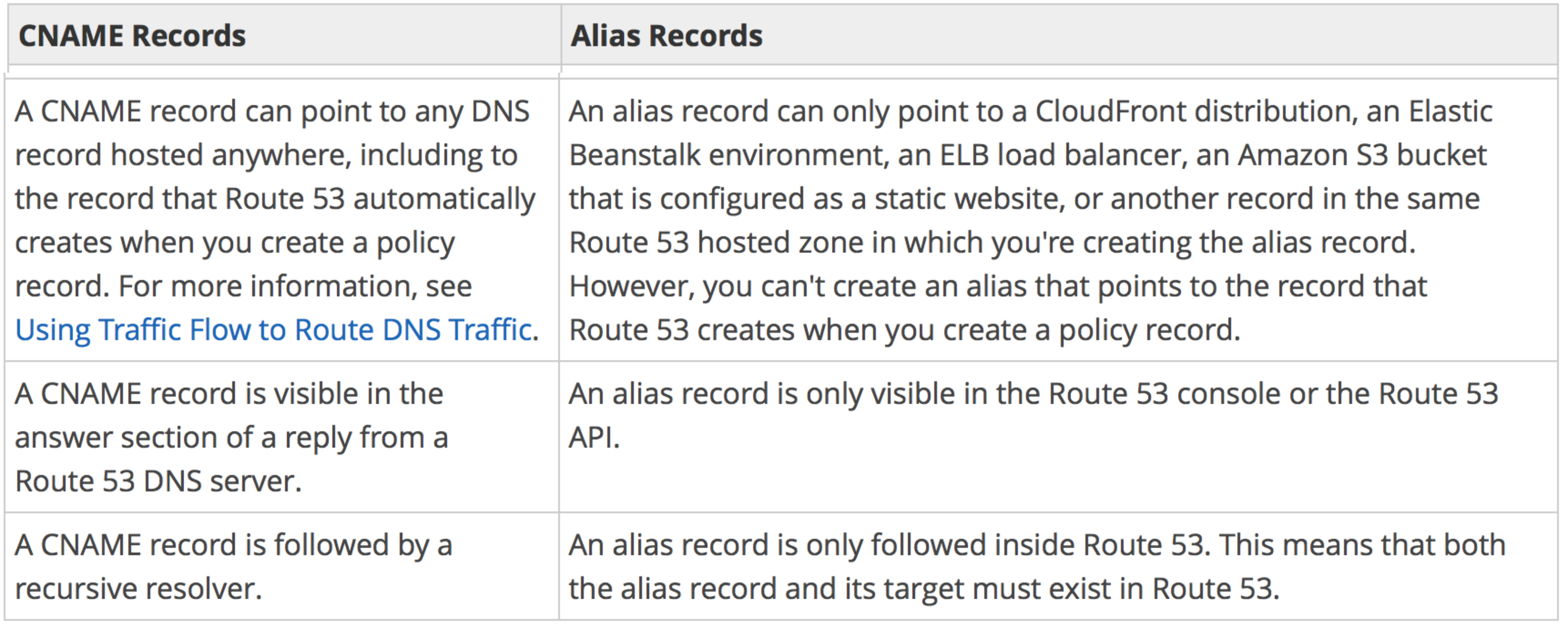


Alias records are similar to CNAME records, but there are some important differences:

CNAME Records	Alias Records
Route 53 charges for CNAME queries.	Route 53 doesn't charge for alias queries to CloudFront distributions, Elastic Beanstalk environments, ELB load balancers, or Amazon S3 buckets. For more information, see Amazon Route 53 Pricing.
You can't create a CNAME record at the top node of a DNS namespace, also known as the *zone apex*. For example, if you register the DNS name example.com, the zone apex is example.com.	You can create an alias record at the zone apex. **Note** If you create an alias record that routes traffic to another record in the same hosted zone, and if the record that you're routing traffic to has a type of CNAME, you can't create an alias record at the zone apex. This is because the alias record must have the same type as the record you're routing traffic to, and creating a CNAME record for the zone apex isn't supported even for an alias record.
A CNAME record redirects queries for a domain name regardless of record type.	Route 53 follows the pointer in an alias record only when the record type also matches.

Amazon ROUTE 53

CNAME vs ALIAS Records



CNAME Records	Alias Records
A CNAME record can point to any DNS record hosted anywhere, including to the record that Route 53 automatically creates when you create a policy record. For more information, see Using Traffic Flow to Route DNS Traffic.	An alias record can only point to a CloudFront distribution, an Elastic Beanstalk environment, an ELB load balancer, an Amazon S3 bucket that is configured as a static website, or another record in the same Route 53 hosted zone in which you're creating the alias record. However, you can't create an alias that points to the record that Route 53 creates when you create a policy record.
A CNAME record is visible in the answer section of a reply from a Route 53 DNS server.	An alias record is only visible in the Route 53 console or the Route 53 API.
A CNAME record is followed by a recursive resolver.	An alias record is only followed inside Route 53. This means that both the alias record and its target must exist in Route 53.

Amazon Route53

Health Checks

Route 53 – Health Checks

- **Route 53 supports** HTTP, HTTPS, TCP health checks

- You can define the IP address or the domain of the endpoint to be monitored by Route 53 Health checks
  - The endpoint can be in AWS or off AWS (charge is higher for off AWS endpoints)
  - Route 53 can't check the health of resources that have an IP address in local, private, non-routable, or multicast ranges.

- Route 53 begins to check the health of the endpoint that you specified in the health check when you associate a health check with the record.

- Optionally, if configured, Route 53 can notify CloudWatch of unhealthy instances, it sets a CloudWatch Alarm, Then CloudWatch will use SNS to send a notification about the unhealthy endpoint.

- If one or more records in a group of records do not have health checks associated with them,
  - Route 53 will treat these records as healthy, since it has no means to decide the health of the corresponding resource(s).

Route 53 – Health Checks

- Create health checks for the resources that you can't create alias records for, this includes EC2 instances or servers in an on-premise data center.
  - **For Alias records** – Best is to specify **Yes** for **Evaluate Target Health** parameter

- Each health check created can monitor one of the following:
  - The health of the specified endpoint/resource, such as a web server
  - The status of other health checks themselves
    - Example, when it is required to be notified if 3 out of 6 available web servers (all are monitored by Route 53) are unhealthy.
  - The status of a configured CloudWatch alarm

- Firewalls, security groups, NACLs..etc, need to have rules configured to allow Route 53 to send regular requests to the endpoints specified in the configured health checks.

Amazon Route53

Routing Policies

AWS ROUTE 53

Choosing a Route 53 Routing Policy

- When you create a record, you choose a routing policy, which determines how Amazon Route 53 responds to queries. Possible values are:

- **Simple routing policy** – Default –
  - Use for a single resource that performs a given function for your domain,
  - Use case: a web server that serves content for the dolfined.com website.

- **Failover routing policy** –
  - Use when you want to configure active-passive failover.

- **Geolocation routing policy** –
  - Use when you want to route traffic based on the location of your users.

AWS ROUTE 53

Choosing a Route 53 Routing Policy

- **Latency routing policy** –
  - Use when you have resources in multiple locations and you want to route traffic to the resource that provides the best latency.
- **Weighted routing policy** –
  - Use to route traffic to multiple resources in proportions that you specify.
- **Geoproximity routing policy** (Requires Route Flow) –
  - Use when you want to route traffic based on the location of your resources and, optionally, shift traffic from resources in one location to resources in another.
- **Multivalue answer routing policy** –
  - Use when you want Route 53 to respond to DNS queries with up to eight healthy records selected at random.

AWS ROUTE 53

Failover Routing

Failover routing lets you route traffic to a resource when the resource is healthy

- If the main resource is not healthy, then route traffic to a different resource
- The primary and secondary records can route traffic to anything from an Amazon S3 bucket that is configured as a website to a complex tree of records.

Amazon ROUTE 53

Geolocation Routing

- Geolocation routing lets you choose the resources that serve your traffic based on the geographic location of your users, i.e the location that DNS queries originate from.

- For example, you may have presence in USA and Europe and want users in the US to be served in the US, and those in Europe to be served by servers in Europe.

- Use cases and benefits for using geolocation routing,
  - Localize your content and present some or all of your website in the language of your users.
  - Use geolocation routing to restrict distribution of content to only the locations in which you have distribution rights.
  - Balancing load across endpoints in a predictable, easy-to-manage way, so that each user location is consistently routed to the same endpoint.

Amazon ROUTE 53

Geolocation Routing

- You can specify geographic locations by continent, by country, or by state in the United States.
- If you create separate records for overlapping geographic regions—for example, one record for North America and one for Canada—priority goes to the smallest geographic region.
  - This allows you to route some queries for a continent to one resource and to route queries for selected countries on that continent to a different resource.
- Geolocation works by mapping IP addresses to locations.
- However, some IP addresses aren't mapped to geographic locations,
  - For those IP addresses, even if you create geolocation records that cover all seven continents, Amazon Route 53 will receive some DNS queries from locations that it can't identify.
  - You can create a default record that handles both queries from IP addresses that aren't mapped to any location and queries that come from locations that you haven't created geolocation records for.
  - If you don't create a default record, Route 53 returns a "no answer" response for queries from those locations.

Amazon ROUTE 53

Latency Based Routing

- If an application is hosted in multiple regions, performance for your users can be improved by serving their requests from the Amazon region that provides the lowest latency.
- To use latency-based routing, you create latency records for your resources in multiple Regions.
- When Amazon Route 53 receives a DNS query for your domain or subdomain,
  - It determines which Amazon regions you've created latency records for,
  - Determines which region gives the user the lowest latency,
  - Then selects a latency record for that region,
  - Route 53 responds with the value from the selected record, such as the IP address for a web server.

Amazon ROUTE 53

Weighted Routing

- Weighted routing lets you associate multiple resources with a single domain name, or subdomain name, and choose how much traffic is routed to each resource.
- This can be useful for a variety of purposes, including load balancing and testing new versions of software.
- To configure weighted routing, you create records that have **the same name and type** for each of your resources.
- You assign each record a relative weight that corresponds with how much traffic you want to send to each resource.
- Amazon Route 53 sends traffic to a resource based on the weight that you assign to the record as a proportion of the total weight for all records in the group:

Weight of the specified Record / Sum of the weight of all records

Amazon ROUTE 53

Geoproximity Routing

- Use Geoproximity routing to have Amazon Route 53 route traffic to your resources based on the geographic location of your users and your resources.
- You can also optionally choose to route more traffic or less to a given resource by specifying a value, known as a *bias*.
  - A bias expands or shrinks the size of the geographic region from which traffic is routed to a resource.
- Route 53 traffic flow is required to use Geoproximity routing.
- To create geoproximity rules for the resources, specify one of the following values for each rule:
  - If you're using AWS resources, the AWS Region that you created the resource in
  - If you're using non-AWS resources, the latitude and longitude of the resource

Traffic Flow

- Route 53 traffic flow provides a visual editor that help in creating complex trees easily.
- The created configuration (routing tree) can be saved as a *traffic policy*
- You can associate the traffic policy with one or more domain names (such as example.com) or subdomain names (such as www.example.com), in the same hosted zone or in multiple hosted zones.
- You can only use traffic flow to create configurations for public hosted zones.

Route 53 – MultiValue Answer

- **Route 53 supports** configuring MultiValue answer routing policy, where more than one IP will be returned in the response to the DNS query
  - It's not a substitute for a load balancer, but the ability to return multiple health-checkable IP addresses is a way to use DNS to improve availability and load balancing.
- It support health checks on the different Route 53 records
  - Which means only healthy endpoint will be returned.
- Route 53 responds to DNS queries with up to eight healthy records and gives different answers to different DNS resolvers.
- If a web server becomes unavailable after a resolver caches a response, client software can try another IP address in the response.
- When all records are unhealthy, Route 53 responds to DNS queries with up to eight unhealthy records.

Route 53 Resolver

Resolving DNS Queries Between VPCs And the On-Premise Network

Route 53 Resolver

- A Route 53 DNS Resolver is there by default when you create a VPC in AWS
  - The default function is to resolve DNS queries within the VPC
    - It answers DNS queries for the VPC domain names (for ELBs, EC2 instances…etc within the VPC)
    - For all other Domain names (not within the VPC, such as Public Domain names on the internet), the Route 53 Resolver will do recursive lookups against public DNS resolvers

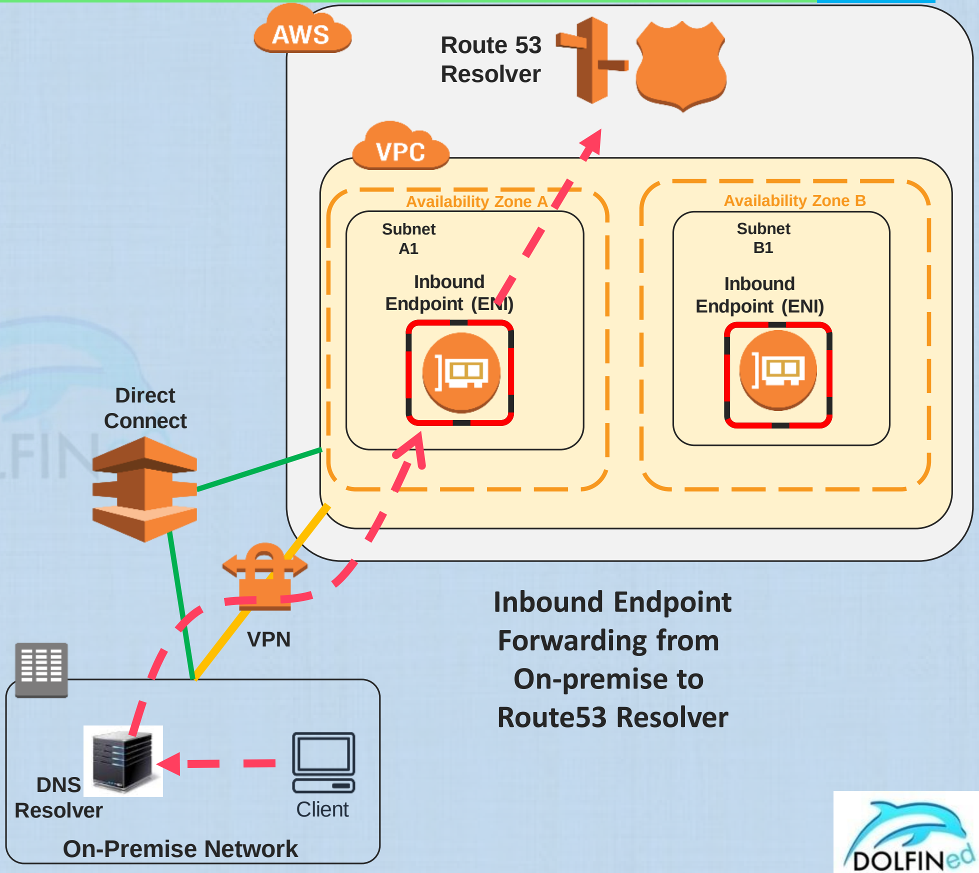
- You can additionally configure (manually) the Route 53 Resolver to:
  - Forward DNS queries from within the VPC to your DNS Resolvers on-premise (Outbound Queries), and/or
  - Answer DNS queries coming from your On-premise network clients (Inbound queries)
    - To allow on-premise clients to use the private hosted zones configured in your VPCs, and
    - Resolve domain names for AWS resources

- The above requires a Direct Connect of a VPN connection between your on-premise network and the AWS VPC(s) in question.

- Route 53 Resolvers are Region specific resources.

Inbound Query Forwarding – On-premise to Route53 Resolver

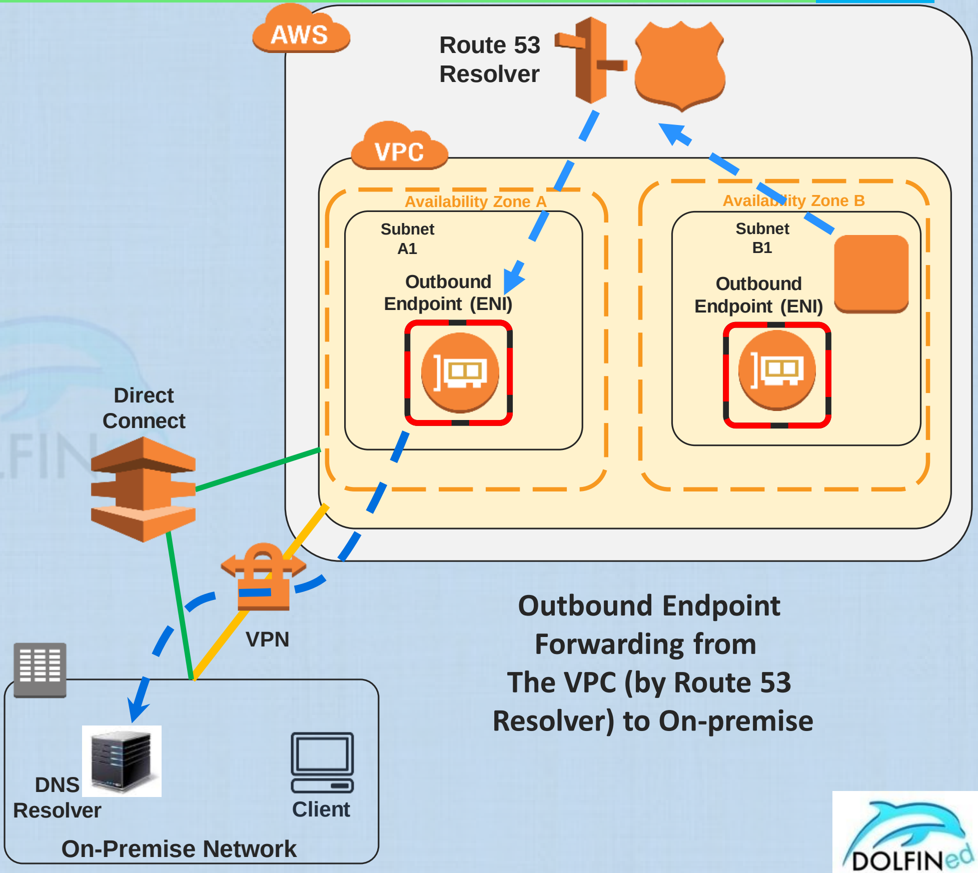
- The inbound endpoint is one more ENIs created in your VPC with the IP address(es) [one ENI for each IP address] that you specified during the endpoint creation. One endpoint is enough to handle inbound direction queries.
- Each endpoint need to be configured with two (min), or more, IP addresses in different subnets and in different availability zones.
  - All in the same AWS Region.
- You can create more than one endpoint, however, For increased load, AWS recommends adding more IP addresses to the same endpoint than creating new endpoints
- On Premise resolvers need to be configured to forward the DNS queries for the domains in AWS to the Inbound endpoints' IP address(es).
- Creating an inbound endpoint doesn't change the behavior of Resolver, it just provides a path from a location outside the AWS network to Resolver.



Inbound Endpoint Forwarding from On-premise to Route53 Resolver

Outbound Query Forwarding – VPC to On-premise DNS Resolver

- To forward DNS queries from EC2 instances in one or more VPCs to the on-premise network, you need to configure outbound endpoints and one or more forwarding rules.
- The outbound endpoint is an ENI created in your VPC with an IP address and a subnet you specified
- The endpoint defines the VPC and IP address that the Resolver will forward DNS queries through (where the queries will originate from)
- You can use the same outbound endpoint for multiple VPCs in the AWS region.
  - Also you can create multiple endpoints
  - Two IP addresses, or more, need to be defined for the endpoint
- A forwarding rule can define a domain name for which the Resolver will forward DNS queries to the on-premise network
- Rules are associated with the VPC(s) for which queries will be forwarded



Outbound Endpoint Forwarding from The VPC (by Route 53 Resolver) to On-premise

Route 53 Resolver - Rules

- Rules are required to help Resolver decide on which DNS queries to forward from VPC(s) to On-premise network (DNS)
- Rules are either Autodefined or Custom Rules
- Rule types:
  - Conditional Forwarding (or Forwarding) Rules
    - They define which DNS queries for which domain(s) to forward to On Premise DNS Resolver
  - System Rules:
    - These will have the Route 53 Resolver respond to queries and not send them to On-premise DNS
  - Recursive Rule:
    - It a rule that gets created automatically by the Route 53 Resolver, it is called Internet Resolver
    - This rules makes Route 53 Resolver act as a recursive resolver for any Domain names that do not have custom rules defined for, and Resolver did not have any auto-defined rules created for them
- Custom rules can be created for AWS domain names, Public domain names, or all domain names.
- Rules are region specific, to use a rule in more than one region, you must create the rule in each region

Sharing Rules between accounts

- You can share the forwarding rules that you created using one AWS account with other AWS accounts.

- To share rules, the Route 53 Resolver console integrates with AWS Resource Access Manager.

- When you create a rule, you specify the outbound endpoint that you want Resolver to use to forward DNS queries to your network.
    - If you share the rule with another AWS account, you also indirectly share the outbound endpoint that you specify in the rule.

- If you used more than one AWS account to create VPCs in an AWS Region, you can do the following:
    - Create one outbound endpoint in the Region.
    - Create rules using one AWS account.
    - Share the rules with all the AWS accounts that created VPCs in the Region.

- This allows you to use one outbound endpoint in a Region to forward DNS queries to your network from multiple VPCs even if the VPCs were created using different AWS accounts.

Amazon Route53

Pricing

AWS ROUTE 53

Route 53 - Pricing

- Hosted Zones
  - $0.50 per hosted zone / month for the first 25 hosted zones
    $0.10 per hosted zone / month for additional hosted zones
  - The monthly hosted zone prices listed above are not prorated for partial months.
  - A hosted zone is charged upon set-up and on the first day of each subsequent month.
  - To allow testing, a hosted zone that is deleted within 12 hours of creation is not charged;
    - However, any queries on that zone will be charged at the rates below.

- Query charges
- Health Check charges
- Traffic flow policy records
- Route 53 Resolver and Recursive DNS Queries

AMAZON GLOBAL ACCELERATOR

Amazon Global Accelerator

Global Accelerator

Without Global Accelerator – Client Issues

- Client IP cache concerns when failures happen

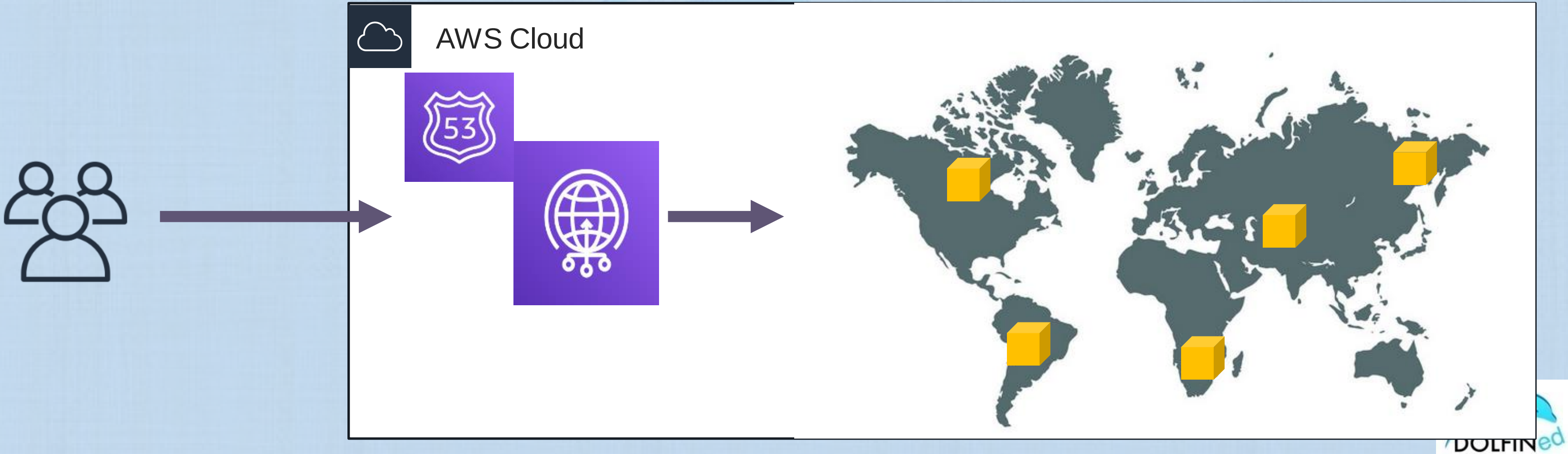
- All traffic is sent over the internet from Client to Applications,
  - Inconsistent performance

- Complex to manage through IP routing configurations on the customer side

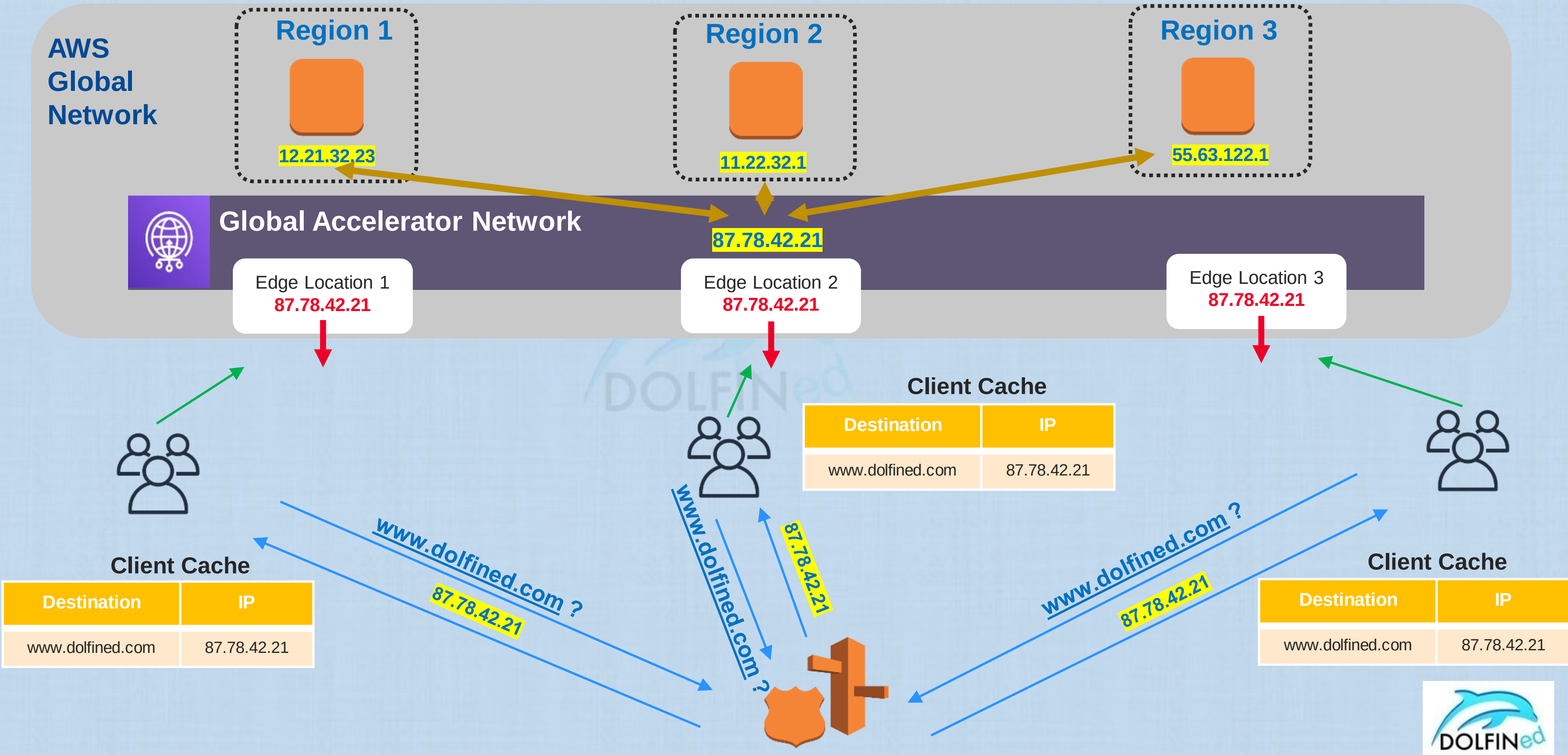
Without Global Accelerator – What is it?

- Global Accelerator is a network layer service that is horizontally scaled and highly available.
- It can be deployed in front of your Internet facing applications
- Incoming user traffic gets distributed intelligently across multiple endpoints in one or more AWS regions
- This will improve applications' availability and performance for the global user base

- With Global Accelerator Traffic from users to applications enters the AWS network at the nearest edge locations, then it is carried over the AWS global network, better security and improved performance/QoS



With Global Accelerator – The Solution



Client Cache

Destination	IP
www.dolfined.com	87.78.42.21

Client Cache

Destination	IP
www.dolfined.com	87.78.42.21

Client Cache

Destination	IP
www.dolfined.com	87.78.42.21

With Global Accelerator

- All Users point to the same static IPs. Anycast IPs advertised from all Gloabal Accelerator locations
- Since all Global Accelerator Locations can serve the content
  - Clients are directed to the closest location then the traffic is carried over the AWS global network
- Endpoints can be EC2 Instances, Application & Network Load Balancers, and Elastic IP addresses
- Global Accelerator maps the Anycast IP(s) to Endpoint IP address.

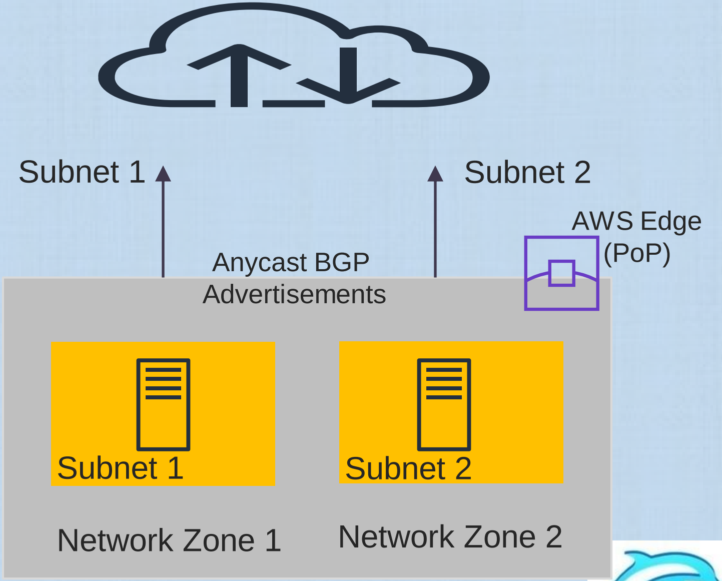
Global Accelerator – Key Benefits

- Two Static Anycast IP addresses are provided for each accelerator created
  - They are assigned to the accelerator as long as it exists, even if it is disabled and not routing traffic
  - They are taken back only when the accelerator is deleted
- Intelligent traffic distribution
- Enhanced fault tolerance to customer applications
- Supports both TCP and UDP protocols
- Endpoint health check monitoring and instant failover to another endpoint when the active endpoint becomes unhealthy
- Traffic rides over AWS's Global Network – Much better performance
- It integrates with AWS Shield to provide DDoS protection to your applications

- Fixed IP address benefits:
  - Application scaling to new AWS Regions or AZs
  - Migration between endpoint types
  - Whitelisting of IP addresses in Security applications
  - Stack upgrades and performance testing

Global Accelerator – Components

- Accelerator
- DNS Name that points to the allocated pair of static IP addresses assigned (Two IPs from 2 subnets)
  - a1234567890abcdef.awsglobalaccelerator.com
- Network Zones
  - Two network zones service the pair of IP addresses assigned to your accelerator
  - They are isolated from one another (think availability zones)
  - If one fails or is blocked, the other can serve the entire traffic for the accelerator

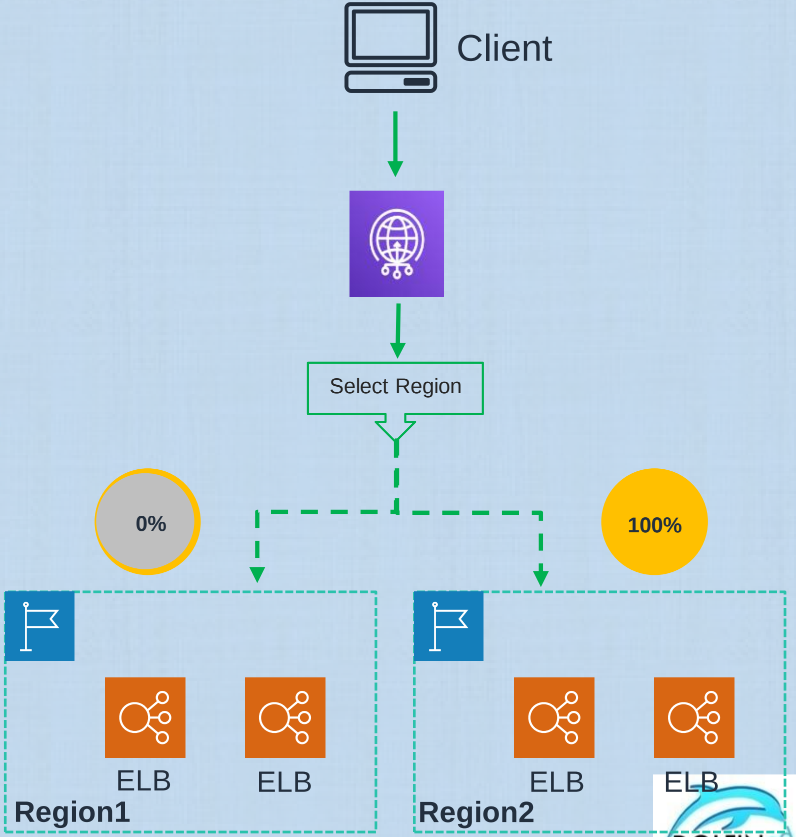


Global Accelerator – Components

- Listeners
  - Is like the listener ports in load balancer, it listens on the configured port or port range for incoming traffic
  - Both TCP and UDP are supported
  - Each listener has one or more Endpoint Groups associated with it
    - Traffic gets forwarded to an endpoint in one of the associated groups
- Endpoint Group
  - Is a collection of one or more Endpoints in a single region
- Endpoints
  - Can be an ALB, NLB, EC2 instance, or Elastic IP addresses.
  - An ALB endpoint can be Internet-facing or Internal

Regional Traffic Dial

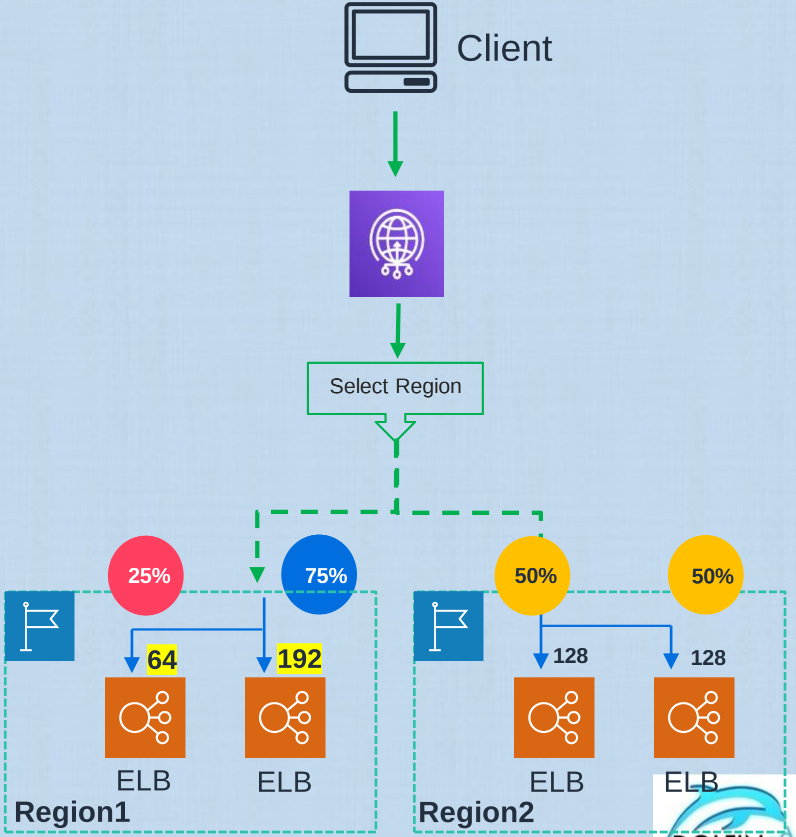
- **Regional Traffic Dial** can be configured at the endpoint group level to increase (dial up) or reduce (dial down) the percentage of traffic that will be accepted (directed) to an endpoint group
  - Useful for performance testing or blue/green deployments



Endpoint Weight

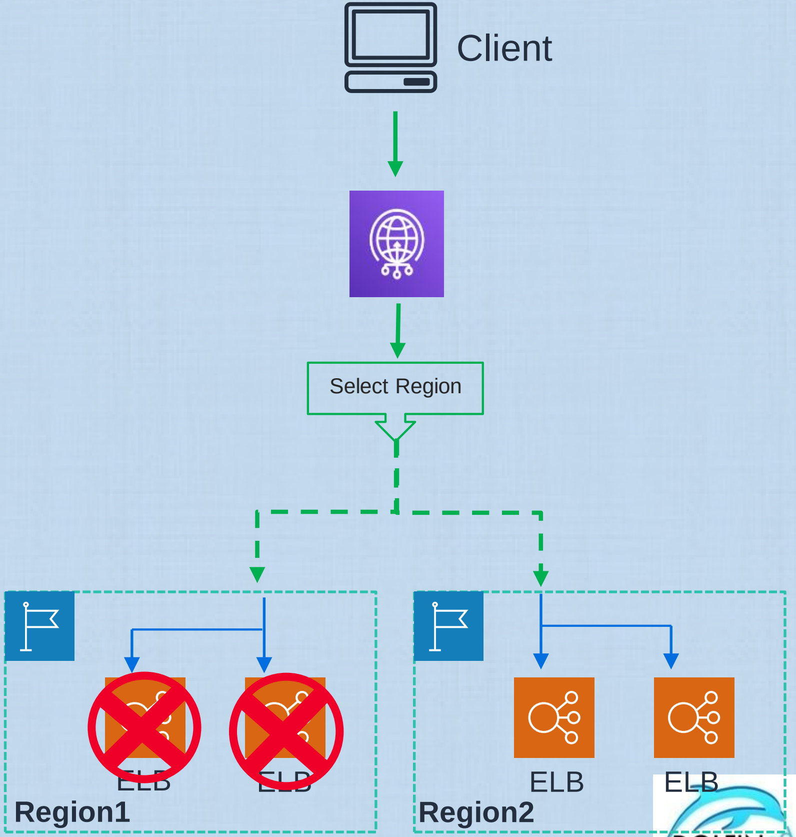
- Weight of an endpoint is a number that you can configure to specify the proportion of traffic to route to the endpoint within an Endpoint group

- Default is 128
- Min is 0
- Max is 256



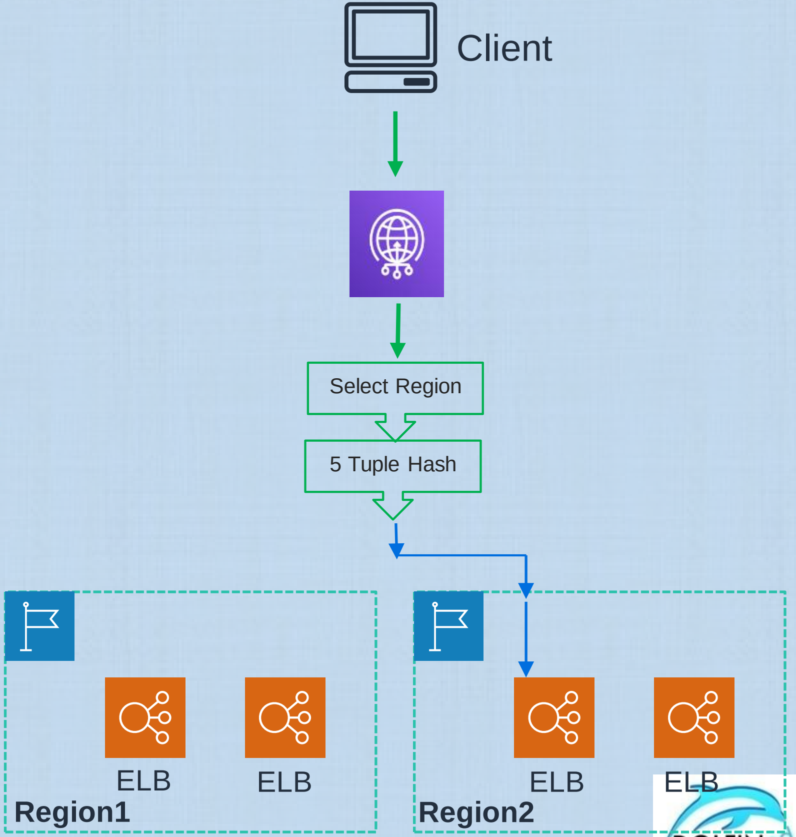
Global Accelerator – Endpoint Health Checks

- AWS Accelerator continuously checks the health of the endpoints associated with the accelerator’s static IPs
- Traffic is directed to healthy endpoints only
- If there are no healthy endpoints the accelerator fails open
- You can configure the health checks at the endpoint group level
  - For ALB/NLB, their configured ELB health checks are used
  - For EC2 and Elastic IP addresses, the configured endpoint group settings are used (TCP/HTTP(s))
  - For UDP listeners, currently Global Accelerator only supports TCP health checks, hence, endpoints must have an active TCP health check process running



Endpoint Selection

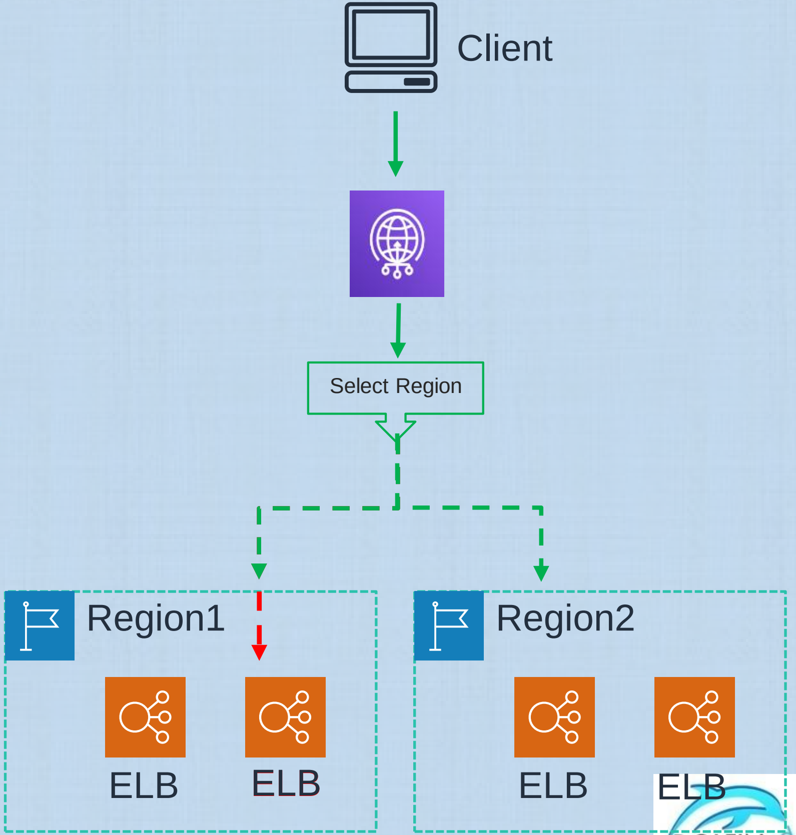
- A region is decided based on Geo-Proximity of the user (client)
- Only regions with healthy endpoints are considered, and within a region only the healthy endpoints are considered
- Endpoint selection is based on a 5 Tuple hashing process where the protocol (TCP/UDP), Source IP and Port, Destination IP and Port are all taken into the hash to decide on the endpoint
  - If client affinity is configured, then the same endpoint will be used to serve the traffic
- Customer configured policies for Traffic Dial and Endpoints weights



Global Accelerator – Client Affinity

- If you have stateful applications, you can choose to have Global Accelerator direct all requests from a user at a specific source (client) IP address to the same endpoint resource, to maintain client affinity.

- When configured, Optimal/Target region is selected, then
  - ○ Traffic is directed to an endpoint which is selected based on a 2-tuple hash that is based on source (client) and endpoint IP address.
  - ○ All traffic from that source IP to the destination is routed to the same endpoint for that flow.



Global Accelerator – Client IP address preservation

- With this feature, you preserve the source IP address of the original client for packets that arrive at the endpoint.

- You can use this feature with Application Load Balancer and EC2 instance endpoints

- When you use an internet-facing Application Load Balancer as an endpoint with Global Accelerator, you can choose to preserve the source IP address of the original client for packets that arrive at the load balancer by enabling client IP address preservation.

- When you use an internal Application Load Balancer or an EC2 instance with Global Accelerator, the endpoint always has client IP address preservation enabled.

Global Accelerator – Use cases

- Application that require whitelisting of a small number of IPs
  - Autonomous vehicles
  - Payment/retail transactions
  - Healthcare
  - IoT
- Multi-region applications:
  - Financial Services
  - DR/Failover scenarios
- UDP traffic applications
  - Gaming
  - Voice over IP
  - DNS
- Live Video Ingestion for media applications
  - Latency sensitive applications

Global Accelerator – Visibility and Monitoring

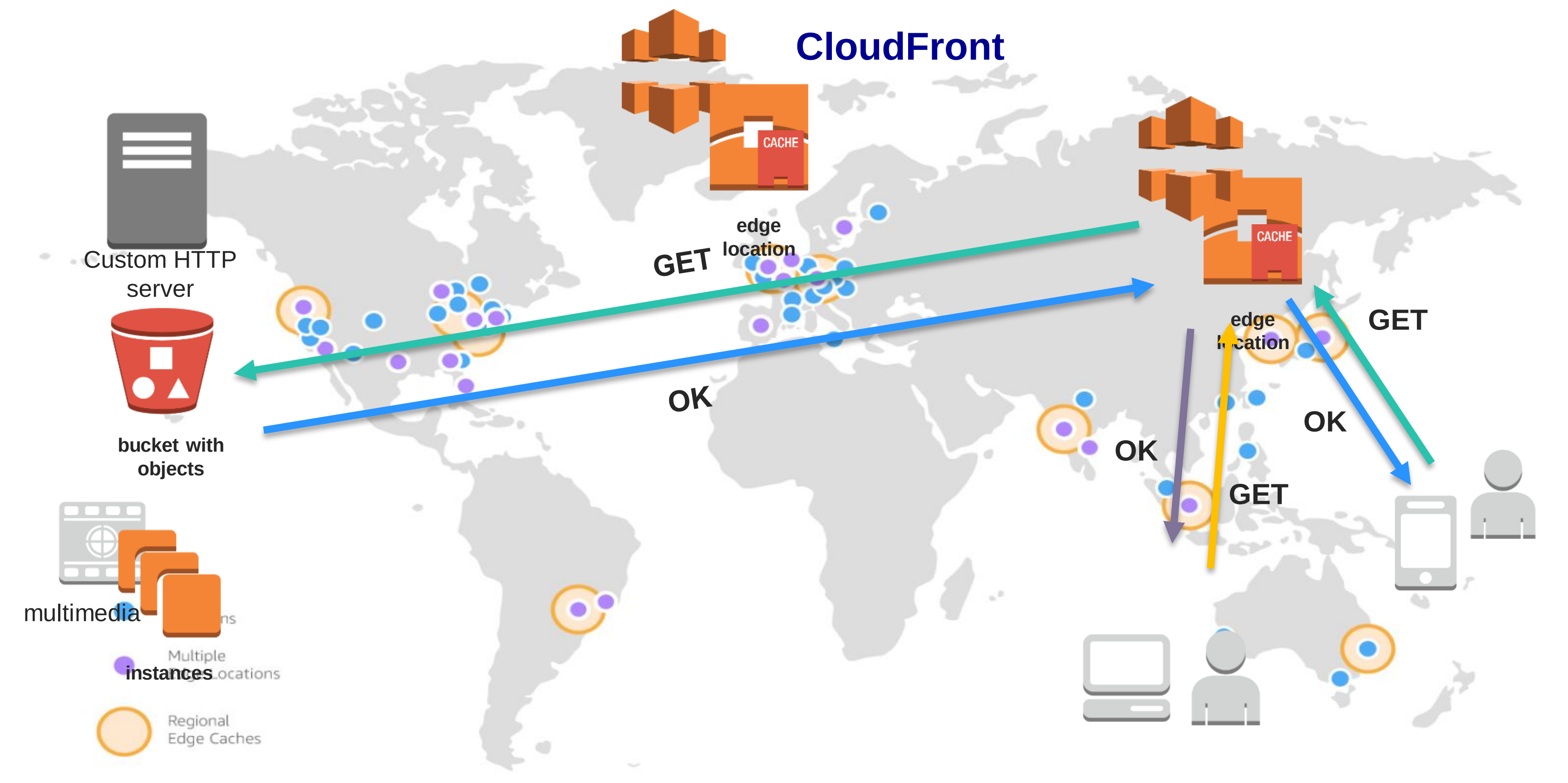
- **Flow Logs** are used to capture information about IP traffic going to and from ENIs in your configured accelerators.
  - Flow log data is published to S3 to a bucket that you can specify.
  - Log data captured includes among other information: Client IP address/port, Endpoint region, Endpoint IP address/port, statistics about packets and bytes.
- **CloudWatch:**
  - Global Accelerators published metrics to AWS CloudWatch, every 60 seconds
    - Only when requests are flowing through the accelerator
- **CloudTrail:**
  - It logs all API calls made to global accelerator APIs

AMAZON CLOUDFRONT

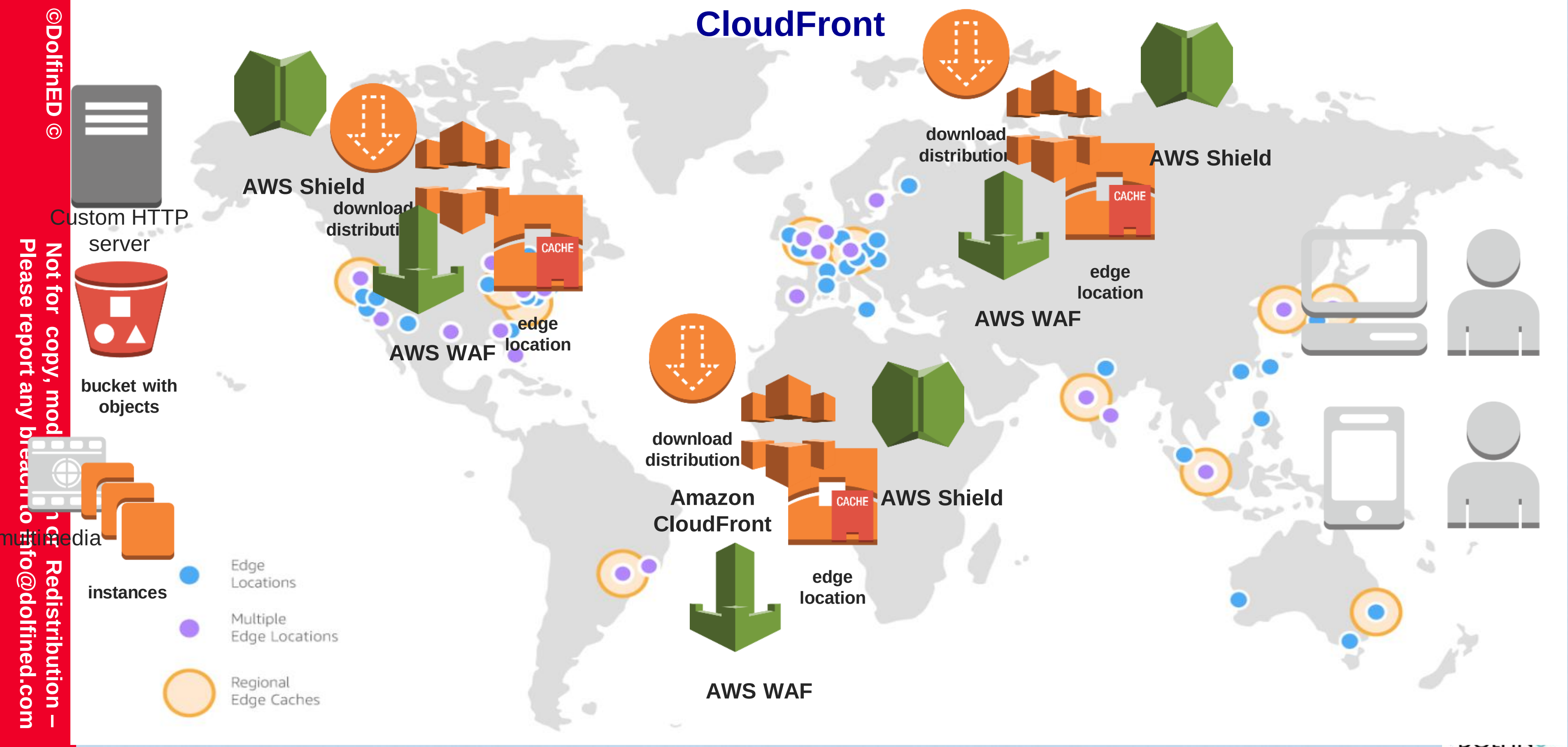
Amazon CloudFront

Introduction

AWS Cloud Front



AWS Cloud Front



CloudFront

Custom HTTP server

bucket with objects

multimedia

instances

AWS Shield

download distribution

edge location

AWS WAF

Amazon CloudFront

Review Topic : CloudFront

Cloudfront

- Cloudfront is a global (not regional) service.
  - It is used Ingress (injection proxy) to upload objects
  - and egress to distribute content

- Amazon Cloudfront is a web service that speeds up distribution of your static and dynamic web content, such as .html, .css, .js, and image files, to your users.

- Cloudfront delivers your content through a worldwide network of data centers called edge locations.

- When a user requests content that you're serving with Cloudfront, the user is routed (via DNS resolution) to the edge location that provides the lowest latency, so that content is delivered with the best possible performance.

Review Topic : CloudFront

Cloudfront – Faster Response to Client requests

- If the content is already in the edge location with the lowest latency, CloudFront delivers it immediately.
  - This dramatically reduces the number of networks that your users' requests must pass through, which improves performance.
- If not, CloudFront retrieves it from an Amazon S3 bucket or an HTTP/web server that you have identified as the source for the definitive version of your content (Origin Server).
- CloudFront also keeps persistent connections with origin servers so files are fetched from the origins as quickly as possible.

Amazon CloudFront

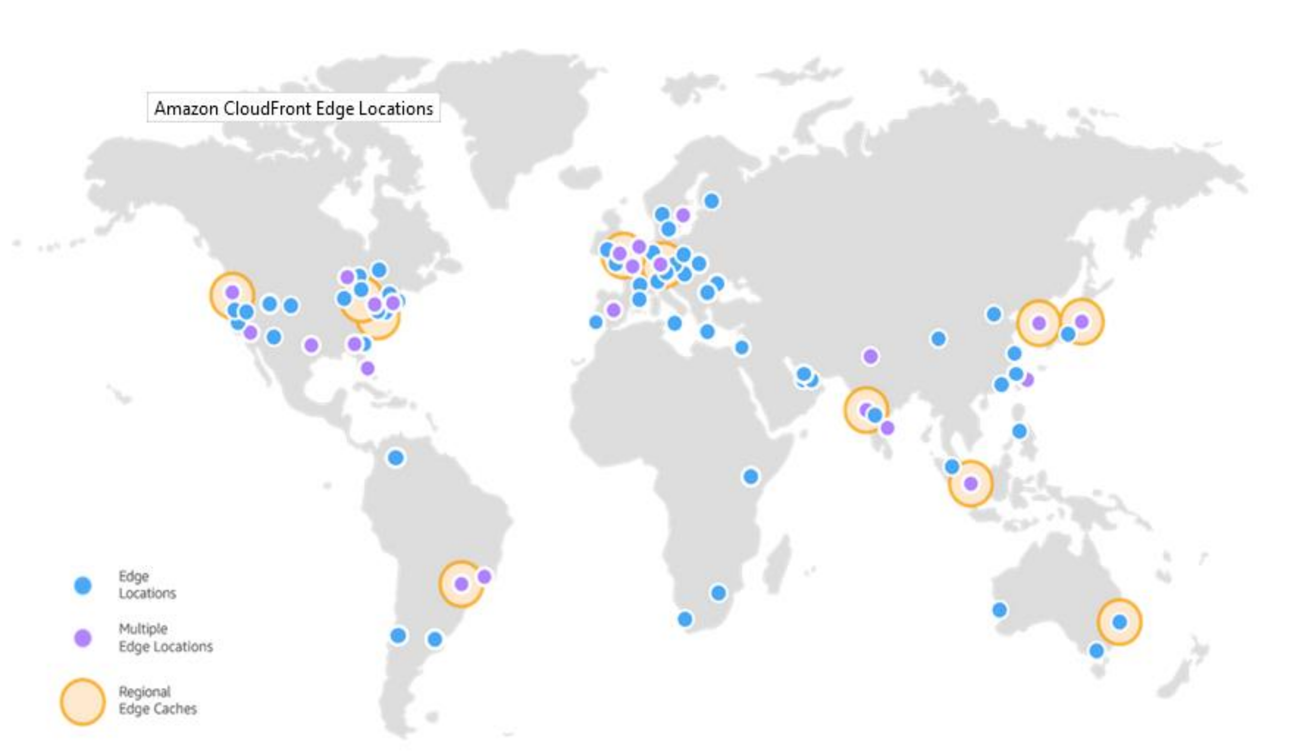
Cache and Regional Edge Cache

Review Topic : CloudFront

CloudFront – Edge Locations

- Edge locations are not tied to Availability Zones, or Regions.

- Amazon Cloudfront has 216 Points of Presence (205 Edge Locations and 11 Regional Edge Caches) in 84 cities across 42 countries.



https://aws.amazon.com/cloudfront/details/

Review Topic : CloudFront

CloudFront- Regional Edge Cache

- Amazon CloudFront has added several **regional edge cache** locations globally, at close proximity to your viewers.
  - They are located between your origin webserver and the global edge locations that serve content directly to your viewers.
  - As objects become less popular, individual edge locations may remove those objects to make room for more popular content.
  - Regional Edge Caches have a larger cache width than any individual edge location, so objects remain in the cache longer at the nearest regional edge caches.
- Regional edge cache locations are currently used only for requests that need to go back to a custom origin; i.e. requests to S3 origins will skip regional edge cache locations.
- Serving less popular content from Regional Edge caches is enabled by default for all new and existing CloudFront distributions. There are no additional charges to use this feature.

Review Topic : CloudFront

CloudFront Regional Edge Cache – How it works

- When a viewer makes a request on your website or through your application, DNS routes the request to the CloudFront edge location that can best serve the user's request.
- This location is typically the nearest CloudFront edge location in terms of latency.
- In the edge location, CloudFront checks its cache for the requested files.
  - If the files are in the cache, CloudFront returns them to the user.
  - If the files are not in the cache, the edge servers go to the nearest regional edge cache to fetch the object.

Amazon CloudFront

Distributions

Review Topic : CloudFront

CloudFront – Configuration Settings

When you want to use CloudFront to distribute your content, you create a distribution and specify configuration settings such as:

- Your origin, which is the Amazon S3 bucket or HTTP server from which CloudFront gets the files that it distributes.

- You can specify any combination of up to 25 Amazon S3 buckets and/or HTTP servers as your origins.

Review Topic : CloudFront

CloudFront Distributions – Web (or Progressive Download) Distribution

- You can use web distributions to serve the following content over HTTP or HTTPS:
  - Static and dynamic download content, for example, .html, .css, .js, and image files, using HTTP or HTTPS.
  - Multimedia content on demand using progressive download and Apple HTTP Live Streaming (HLS).
- **You can't** serve Adobe Flash multimedia content over HTTP or HTTPS, but you can serve it using a CloudFront RTMP distribution.
- For web distributions, your origin can be either an Amazon S3 bucket or an HTTP server
- A live event, such as a meeting, conference, or concert, in real time.
  - For live streaming, you create the distribution automatically by using an AWS CloudFormation stack.

CloudFront Distribution – RTMP (or Streaming) Distribution

- RTMP distributions stream media files using Adobe Media Server and the Adobe Real-Time Messaging Protocol (RTMP).
- An RTMP distribution must use an Amazon S3 bucket as the origin.
- CloudFront lets you create a total of up to 200 web distributions and 100 RTMP distributions for an AWS account.

Review Topic : CloudFront

CloudFront Distributions – Configuration Propagation

When you save changes to your distribution configuration,

- CloudFront starts to propagate the changes to all edge locations.
- Until your configuration is updated in an edge location, CloudFront continues to serve your content from that location based on the previous configuration.
- Your changes don't propagate to every edge location instantaneously.
  - While CloudFront is propagating your changes, AWS can't determine whether a given edge location is serving your content based on the previous configuration or the new configuration.
  - When propagation is complete, the status of your distribution changes from **InProgress** to **Deployed**.
- After your configuration is updated in an edge location, CloudFront immediately starts to serve your content from that location based on the new configuration.

Amazon CloudFront

Origin and Custom Origin Servers

Review Topic : CloudFront

Origin Servers

- You specify *origin servers*, like an Amazon S3 bucket or your own HTTP server, from which CloudFront gets your files.
- An origin is the location where you store the original definitive version of your web content, which you want to distribute via Cloudfront.
  - If you're serving content over HTTP, your origin server is either an Amazon S3 bucket or an HTTP server, such as a web server.
  - Your HTTP server can run on an Amazon Elastic Compute Cloud (Amazon EC2) instance or on a server that you manage; these servers are also known as *custom origins.*
  - If you use the Adobe Media Server RTMP protocol to distribute media files on demand, your origin server is always an Amazon S3 bucket.
- Only S3 buckets are considered as Origins
  - S3 buckets configured as static website hosting are Custom Origins, same for EC2 instances and ELBs as origins.

Review Topic : CloudFront

Using S3 bucket as a CloudFront Origin

Using Amazon S3 Buckets for Your Origin

- When you use Amazon S3 as an origin for your distribution, you place any objects that you want CloudFront to deliver in an Amazon S3 bucket.

- You can use any method that is supported by Amazon S3 to get your objects into Amazon S3 (the Amazon S3 console or API, or a third-party tool).

- **You can create a hierarchy in your bucket to store the objects, just as you would with any other Amazon S3 bucket.**

- Using an existing Amazon S3 bucket as your CloudFront origin server doesn't change the bucket in any way;
  - You can still use it as you normally would to store and access Amazon S3 objects at the standard Amazon S3 price.

Review Topic : CloudFront

Using EC2 Instance as a CloudFront Custom Origin Server

- A custom origin is an HTTP server, the HTTP server can be an Amazon EC2 instance, an ELB, or an HTTP server that you manage privately.

- When you use a custom origin that is your own HTTP server, you specify the DNS name of the server, along with the HTTP and HTTPS ports and the protocol that you want CloudFront to use when fetching objects from your origin.

- Most CloudFront features are supported when you use a custom origin with the following exceptions:
  - **RTMP distributions**—Not supported (the origin must be an S3 Bucket for Media files).
  - **Private content**—Although you can use a signed URL to distribute content from a custom origin, for CloudFront to access the custom origin, the origin must remain publicly accessible.

Review Topic : CloudFront

Using EC2 Webserver as a CloudFront Custom Origin Server

If you use Amazon Elastic Compute Cloud for your custom origins, AWS recommends the following:

- Use an Amazon Machine Image that automatically installs the software for a web server.
- Use an Elastic Load Balancing load balancer to handle traffic across multiple Amazon EC2 instances and to isolate your application from changes to Amazon EC2 instances.
- When you create your CloudFront distribution, specify the URL of the load balancer for the domain name of your origin server.

Review Topic : CloudFront

Using S3 bucket' static website as CloudFront Custom Origin Server

- You can set up an Amazon S3 bucket that is configured as a website endpoint **as custom origin with CloudFront.**

- When you specify the bucket name in this format as your origin, you can use Amazon S3 redirects and Amazon S3 custom error documents.

Amazon CloudFront

Cache Behavior

Review Topic : CloudFront

CloudFront Cache Behavior

Cache Behavior

- A cache behavior lets you configure a variety of CloudFront functionality for a given URL path pattern for files on your website.

- For simplicity, the cache behaviors can be seen as routing requests to the correct origins.

  - For instance, a cache behavior might apply to all .gif files in the images directory on a web server that you're using as the origin server in CloudFront.

- List the cache behaviors in the order that you want CloudFront to evaluate them in, default will always be the last to be processed

Review Topic : CloudFront

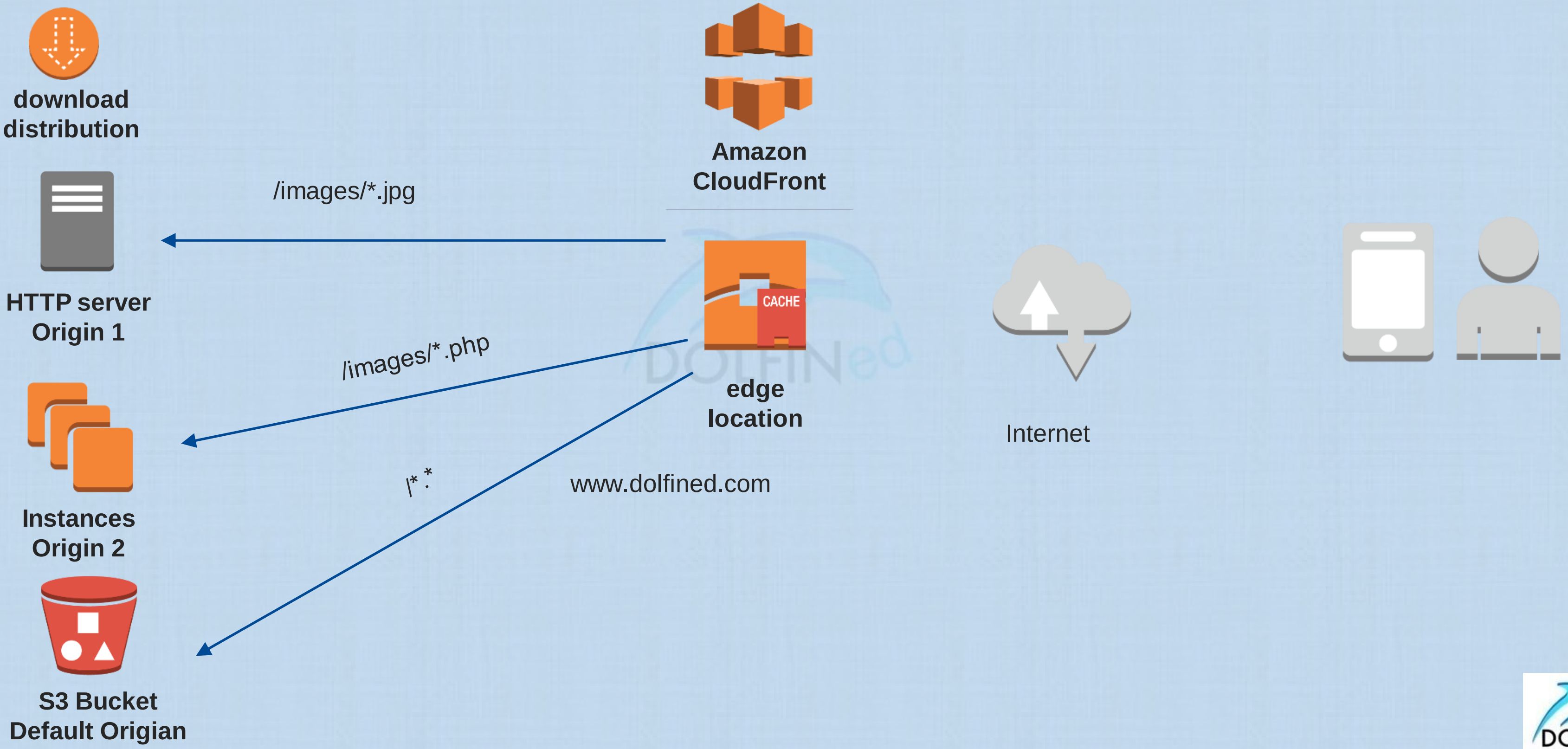
CloudFront Cache Behavior

For each Cache behavior you can configure the following functionality:

- The path pattern.
- If you have configured multiple origins for your CloudFront distribution, which origin you want CloudFront to forward your requests to.
- Whether to forward query strings to your origin.
- Whether accessing the specified files requires signed URLs.
- Allowed HTTP methods, Whether to require users to use HTTPS to access those files.
- The minimum amount of time that those files stay in the CloudFront cache regardless of the value of any Cache-Control headers that your origin adds to the files.

AWS Cloud Front

CloudFront Cache Behavior – Path Pattern & Origin Selection



Review Topic : CloudFront

CloudFront – Path Pattern

Path Pattern

- A path pattern (for example, images/*.jpg) specifies which requests you want this cache behavior to apply to.
- When CloudFront receives an end-user request, the requested path is compared with path patterns in the order in which cache behaviors are listed in the distribution.
  - The first match determines which cache behavior is applied to that request.
  - Example, suppose you have three cache behaviors with the following three path patterns, in this order: images/*.jpg , images/* , *.gif
    - A request for the file images/sample.gif doesn't satisfy the first path pattern, so the associated cache behaviors are not be applied to the request.
    - The file does satisfy the 2nd path pattern, so the cache behaviors associated with the second path pattern are applied even though the request also matches the 3rd path pattern.
- When you create a new distribution, the value of **Path Pattern** for the default cache behavior is set to * (all files) and cannot be changed.
  - This value causes CloudFront to forward all requests for your objects to the origin that you specified in the Origin Domain Name field.

Amazon CloudFront

- Time To Live (TTL)
- Allowed HTTP Methods

Review Topic : CloudFront

CloudFront – Other configuration settings

TTL

- The amount of time, in seconds, that you want objects to stay in CloudFront caches before CloudFront forwards another request to your origin to determine whether the object has been updated.
  - The default value for **Minimum TTL** is 0 seconds, it means never cache any object.
  - The default value for **Maximum TTL** is 31536000 seconds (one year).
  - The default value for **Default TTL** is 86400 seconds (one day).

CloudFront Cache Behavior – HTTP Methods

Allowed HTTP Methods

Specify the HTTP methods that you want CloudFront to process and forward to your origin:

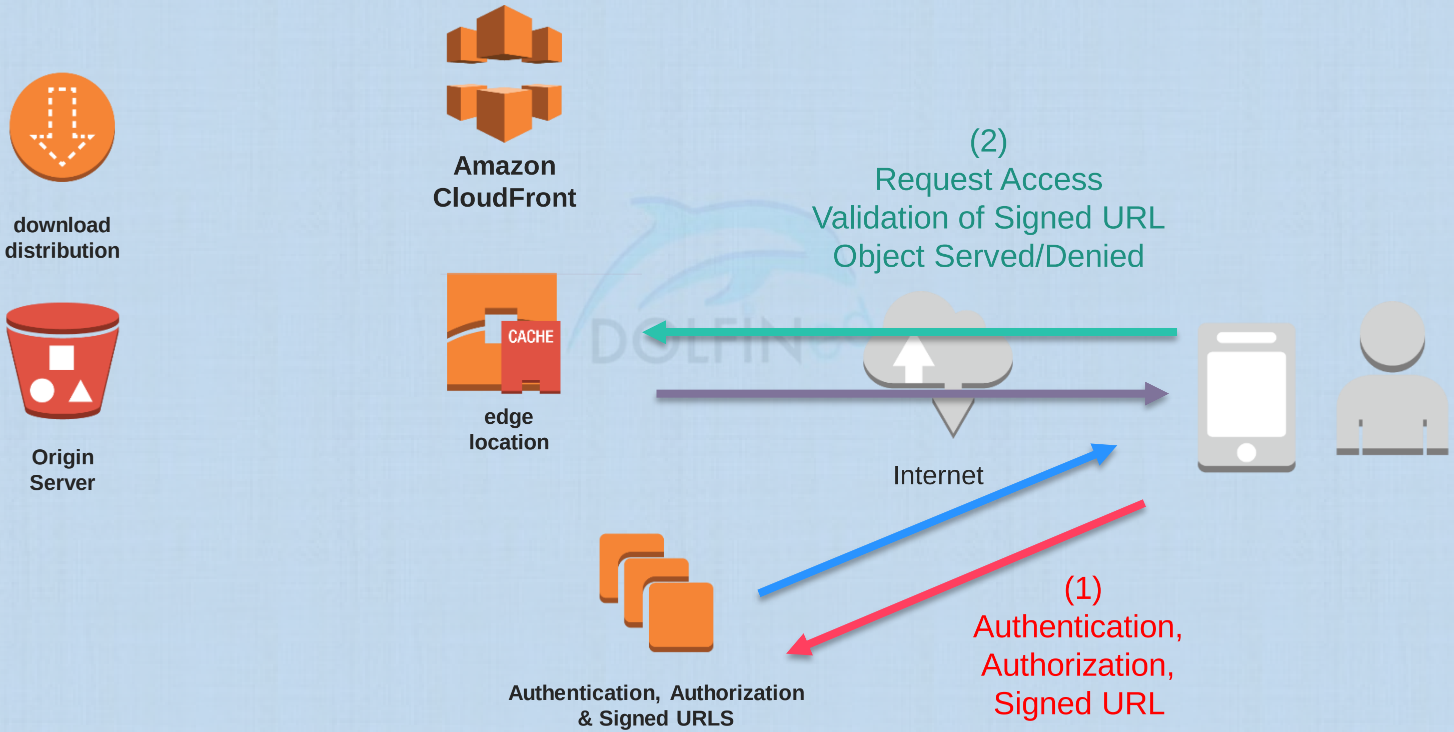
- **GET, HEAD (Cached):** You can use CloudFront only to get objects from your origin or to get object headers.
  - Responses to both methods are cached by default
- **GET, HEAD, OPTIONS:**
  - You can use CloudFront only to get objects from your origin, get object headers, or retrieve a list of the options that your origin server supports.
  - Responses to OPTIONS can be optionally cached
- **GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE:**
  - You can use CloudFront to get, add, update, and delete objects, and to get object headers. In addition, you can perform other POST operations such as submitting data from a web form.
  - Responses to PUT, POST, PATCH, and DELETE can not be cached

Amazon CloudFront

Servings Private Content

AWS Cloud Front

Serving Private Content



Review Topic : CloudFront

Service Private Content via CloudFront

- Companies that distribute content via the internet may need to restrict access to private documents, sensitive business data, internal/subscribed-to media streams, or specific content that is intended for selected users/subscribers,

- An example, users who have paid a fee to subscribe to come content, to securely serve this private content using CloudFront, you can do the following:
  - Require that these users access the private content by using special CloudFront signed URLs or signed cookies.
  - Require that these users access the Amazon S3 content using CloudFront URLs, not Amazon S3 URLs.
    - Restricting access to CloudFront URLs (and S3 URLs) isn't required, but we recommend it to prevent users from bypassing the restrictions that you specify in signed URLs or signed cookies.

Review Topic : CloudFront

CloudFront – Controlling Access to your content

- You can control user access to your private content in two ways, as shown in the following illustration:
  - Restrict access to objects in CloudFront edge caches
  - Restrict access to objects in your Amazon S3 bucket

Review Topic : CloudFront

Restricting Access to Objects in CloudFront Edge Caches

- You can configure CloudFront to require that users access your objects using either **signed URLs** or **signed cookies**.
- Then develop your application either to create and distribute signed URLs to authenticated users OR to send Set-Cookie headers that set signed cookies on the viewers for authenticated users.
  - To give a few users long-term access to a limited number of objects, you can also create signed URLs manually.
- When you create signed URLs or signed cookies to control access to your objects, you can specify the following restrictions:
  - An ending date and time, after which the URL is no longer valid.
  - (Optional) The date and time that the URL becomes valid.
  - (Optional) The IP address or range of addresses of the computers that can be used to access your content.

Review Topic : CloudFront

CloudFront – Trusted Signers

- To create signed URLs or signed cookies, you need at least one AWS account that has an active CloudFront key pair.
- Signed URLS have an expiry attached to them, great way to provide temporary access.
- **Web distributions (both signed URLs and signed Cookies)** –
  - Users can use signed URLs or signed Cookies
- **RTMP distributions (signed URLs only)** – You add trusted signers to a distribution.
  - After you add trusted signers to an RTMP distribution, users must use signed URLs to access any object associated with the distribution.

CloudFront

How Signed URLs Work

- In your CloudFront distribution, specify one or more trusted signers, which are the AWS accounts that you want to have permission to create signed URLs.
- Develop your application to determine whether a user should have access to your content and to create signed URLs for the objects or parts of your application that you want to restrict access to.
- A user requests an object for which you want to require signed URLs.
- Your application verifies that the user is entitled to access the object: they've signed in, they've paid for access to the content, or they've met some other requirement for access.
- Your application creates and returns a signed URL to the user.
  - The signed URL allows the user to download or stream the content.

Amazon CloudFront

Restricting Access to your Origin Servers

CloudFront – Origin Access Identity

To ensure that users can access objects using only CloudFront URLs, regardless of whether the URLs are signed, perform the following tasks:

- Create an origin access identity, which is a special CloudFront user, and associate the origin access identity with your distribution.
  - For web distributions, associate the origin access identity with origins, to secure all or just some of your Amazon S3 content.
  - Also, it is possible to create an origin access identity and add it to your distribution when you create the distribution.
- Change the permissions either on your Amazon S3 bucket or on the objects in your bucket so only the origin access identity has read permission (or read and download permission).
- When users access the Amazon S3 objects through CloudFront, the CloudFront origin access identity gets the objects on behalf of your users.

Review Topic : CloudFront

Restricting Access to Objects in Amazon S3 Buckets

- You can optionally secure the content in your Amazon S3 bucket so users can access it through CloudFront but cannot access it directly by using Amazon S3 URLs.

- To require that users access your content through CloudFront URLs, you perform the following tasks:
  - Create a special CloudFront user called an **origin access identity**.
  - Give the origin access identity S3 bucket policy permission to read the objects in your bucket.
  - Remove permission for anyone else to use Amazon S3 URLs to read the objects.

- This prevents anyone from bypassing CloudFront and using the Amazon S3 URL to get content that you want to restrict access to.
  - This step isn't required to use signed URLs, but AWS recommends it.

Review Topic : CloudFront

Serving Private Content

1) To configure CloudFront to serve private content, perform the following tasks:

- (Optional but recommended) Require your users to access your content only through CloudFront.
- The method that you use depends on whether you're using Amazon S3 or custom origins:
  - **Amazon S3** – Use Origin Access Identity to restrict access to your AWS S3 content
  - **Custom origin** – Use custom headers to restrict access to your content on the custom origin

2) Specify the AWS accounts that you want to use to create signed URLs or signed cookies.

3) Write your application to respond to requests from authorized users either with signed URLs or with Set-Cookie headers that set signed cookies.

Amazon CloudFront

Alternate Domain Names

Review Topic : CloudFront

CloudFront Domain Name vs CNAMEs

Adding and Moving Alternate Domain Names (CNAMEs)

- When you create a distribution, CloudFront returns a domain name for the distribution,
  - as an example: d112211abcdef8.cloudfront.net

- To use a different domain name, such as www.dolfined.com, instead of the cloudfront.net domain name that CloudFront had assigned to your distribution,
  - You can add an alternate domain name to your distribution for www.dolfined.com. You can then use the following URL for /images/tree1.jpg:
  - http://www.dolfined.com/images/tree1.jpg
  - Both web and RTMP distributions support alternate domain names.

Review Topic : CloudFront

CloudFront - Alternate Domain Names

Configure the DNS service for the domain to route traffic for the domain, such as dolfined.com, to the CloudFront domain name for your distribution, such as d112211abcdef8.cloudfront.net.

- The method that you use depends on whether you're using Route 53 as the DNS service provider for the domain:
  - **Route 53** Creates an alias resource record set.
    - With an alias resource record set, you don't pay for Route 53 queries.
    - You can create an alias resource record set for the root domain name (dolfined.com), which DNS doesn't allow for CNAMEs.
  - **Another DNS service provider** Use the method provided by your DNS service provider to add a CNAME resource record set to the hosted zone for your domain.

Amazon CloudFront

Viewer and Origin Protocol Policies

Review Topic : CloudFront

CloudFront Cache Behavior – Viewer Protocol Policy

Viewer Protocol Policy

- Choose the protocol policy that you want viewers to use to access your content in CloudFront edge locations:
  - **HTTP and HTTPS**: Viewers can use both protocols.
  - **Redirect HTTP to HTTPS**: Viewers can use both protocols, but HTTP requests are automatically redirected to HTTPS requests.
  - **HTTPS Only**: Viewers can only access your content if they're using HTTPS.

- For web distributions, you can configure CloudFront to require that viewers use HTTPS to request your objects, so connections are encrypted when CloudFront communicates with viewers.

Review Topic : CloudFront

CloudFront – HTTP to HTTPS Redirects

- Viewers can use both protocols.
  - HTTP GET and HEAD requests are automatically redirected to HTTPS requests.
  - CloudFront returns HTTP status code 301 (Moved Permanently) along with the new HTTPS URL.
  - The viewer then resubmits the request to CloudFront using the HTTPS URL.

AWS Cloud Front

CloudFront – Origin Protocol Policy for Custom Origins

- The protocol policy that you want CloudFront to use when fetching objects from your origin server.

- You can choose one of the following values:
  - **HTTP Only:**
    - CloudFront uses only HTTP to access the origin.
  - **HTTPS Only:**
    - CloudFront uses only HTTPS to access the origin.
  - **Match Viewer:**
    - CloudFront communicates with your origin using HTTP or HTTPS, depending on the protocol of the viewer request.

Note:

- If your Amazon S3 bucket is configured as a website endpoint, you must specify **HTTP Only**
  - Amazon S3 doesn't support HTTPS connections in that configuration.

Amazon CloudFront

Working with Cached Objects – Invalidations

Review Topic : CloudFront

Invalidating Objects (Web Distributions Only)

If you need to remove an object from CloudFront edge caches before it expires, you can do one of the following:

- Invalidate the object from edge caches. The next time a viewer requests the object, CloudFront returns to the origin to fetch the latest version of the object.

- Use object versioning to serve a different version of the object that has a different name.

- You can't cancel an invalidation after you submit it.

Important

- You can invalidate most types of objects that are served by a web distribution, However,
  - You cannot invalidate media files in the Microsoft Smooth Streaming format when you have enabled Smooth Streaming for the corresponding cache behavior.
  - In addition, you cannot invalidate objects that are served by an RTMP distribution.

Amazon CloudFront

Using Web Application Firewalls (WAF)

Review Topic : CloudFront

Using AWS WAF to Control Access to Your Content

- AWS WAF is a web application firewall that lets you monitor the HTTP and HTTPS requests that are forwarded to CloudFront, and lets you control access to your content.

- Based on conditions that you specify, such as the IP addresses that requests originate from or the values of query strings, CloudFront responds to requests either with the requested content or with an HTTP 403 status code (Forbidden).

- You can also configure CloudFront to return a custom error page when a request is blocked.

- After you create an AWS WAF web access control list (web ACL), you create or update a web distribution and associate the distribution with a web ACL. You can associate as many CloudFront distributions as you want with the same web ACL or with different web ACLs.

Amazon CloudFront

Geo Restrictions

Review Topic : CloudFront

Restricting the Geographic Distribution of Your Content

- You can use *geo restriction*, also known as *geoblocking*, to prevent users in specific geographic locations from accessing content that you're distributing through a CloudFront web distribution.

- To use geo restriction, you have two options:
  - Use the CloudFront geo restriction feature.
    - Use this option to restrict access to all of the files that are associated with a distribution and to restrict access at the country level.
  - Use a third-party geolocation service.
    - Use this option to restrict access to a subset of the files that are associated with a distribution or to restrict access at a finer granularity than the country level.

Review Topic : CloudFront

Using CloudFront Geo Restriction

- When a user requests your content, CloudFront typically serves the requested content regardless of where the user is located.

- If you need to prevent users in specific countries from accessing your content, you can use the CloudFront geo restriction feature to do one of the following:
  - Allow your users to access your content only if they're in one of the countries on a **whitelist** of approved countries.
  - Prevent your users from accessing your content if they're in one of the countries on a **blacklist** of banned countries.

Note

- CloudFront determines the location of your users by using a third-party GeoIP database.
- The accuracy of the mapping between IP addresses and countries varies by region is 99.8%
- Be aware that if CloudFront can't determine a user's location, CloudFront will serve the content that the user has requested.

Review Topic : CloudFront

Using a Third-Party Geolocation Service

- The CloudFront geo restriction feature lets you control distribution of your content at the country level for all files that you're distributing with a given web distribution.

- If you have geographic restrictions on where your content can be distributed and the restrictions don't follow country boundaries, or if you want to limit access to only some of the files that you're distributing through CloudFront,
  - You can combine CloudFront with a third-party geolocation service.
  - This can allow you to control access to your content based not only on country but also based on city, zip or postal code, or even latitude and longitude.

- When you're using a third-party geolocation service, we recommend that you use CloudFront signed URLs, which let you specify an expiration date and time after which the URL is no longer valid.
  - In addition, we recommend that you use an S3 bucket as your origin because you can then use a CloudFront origin access identity to prevent users from accessing the S3 content directly

Amazon CloudFront

Video Streaming with CloudFront Web Distributions

Review Topic : CloudFront

Configuring On-Demand Streaming Web Distributions

- You can use CloudFront web distributions to serve **on-demand streaming media files** from any **HTTP origin (i.e Web Distribution)**. Below are several examples of working with different origins to serve streaming video content.

- Configuring On-demand with AWS Elemental Media-Store
- Configuring On-Demand Smooth Streaming
- Configuring On-Demand Progressive Downloads
- Configuring On-Demand Apple HTTP Live Streaming (HLS)

- The key message is, you do not have to use RTMP distributions to serve streaming video using Cloudfront, you can do so using Web distributions.

- **You can't** serve Adobe Flash multimedia content over HTTP or HTTPS, but you can serve it using a CloudFront RTMP distribution.

Review Topic : CloudFront

Serving Media Content by Using HTTP

- When you use HTTP to serve media content, AWS recommends that you use an HTTP-based (i.e web distribution not RTMP) dynamic streaming protocol such as
  - Apple HTTP Dynamic Streaming (Apple HDS),
  - Apple HTTP Live Streaming (Apple HLS), <<- Widely supported
  - Microsoft Smooth Streaming,
  - or MPEG-DASH.
- For dynamic-streaming protocols, a video is divided into a lot of small segments that are typically just a few seconds long each.
  - If the users commonly stop watching before the end of a video (for example, because they close their viewer during the credits),
    - CloudFront has still cached all of the small segments up to that point in the video.

Review Topic : CloudFront

CloudFront – Common Media Streaming Servers

You can use

Wowza Streaming Server

Microsoft IIS media Server

Adobe Media Server, and

As Origins to satisfy your on-demand and/or live streaming requirements with CloudFront

Review Topic : CloudFront

Configuring On-Demand with AWS Elemental MediaStore

- If you store on-demand videos in AWS Elemental MediaStore, you can create a CloudFront distribution to serve the content.
- To get started, you grant CloudFront access to your AWS Elemental MediaStore container.
- Then you create a CloudFront distribution and configure it to work with AWS Elemental MediaStore.
- https://aws.amazon.com/blogs/aws/aws-media-services-process-store-and-monetize-cloud-based-video/

Amazon CloudFront

Video Streaming with CloudFront RTMP Distributions

Review Topic : CloudFront

CloudFront - How RTMP Distributions Work

- To stream media files using CloudFront, you need to provide two types of files to the end users:
  - The media files
  - A media player, for example, **JW Player, Flowplayer, or Adobe Flash**

- End users view the media files using the media player that is provided for them;
  - They do not use the media player (if any) that is already installed on their computer or other device.

- When an end user streams the media file, the media player begins to play the content of the file while the file is still being downloaded from CloudFront.
  - The media file is not stored locally on the end user's system.

Review Topic : CloudFront

How RTMP Distributions Work

- To use CloudFront to serve both the media player and the media files, you need two types of distributions:
  - A web distribution for the media player, and
  - An RTMP distribution for the media files.

- **Web distributions serve files over HTTP**, while **RTMP distributions stream media files over RTMP** (or a variant of RTMP).

- Media files MUST be stored in an AWS S3 bucket, custom origins are not supported

- The media player can be in the same S3 bucket, a different S3 bucket, or in a custom HTTP origin server while served using Cloudfront

- For CloudFront to distribute media files, CloudFront uses **Adobe Flash Media Server as the streaming server** and streams media files **using Adobe's Real-Time Messaging Protocol (RTMP)**.

Amazon CloudFront

Access Logs and Reports

Review Topic : CloudFront

CloudFront - Access Logs

- You can configure CloudFront to create log files that contain detailed information about every user request that CloudFront receives.
- These access logs are available for both web and RTMP distributions.
- If you enable logging, you can also specify the Amazon S3 bucket that you want CloudFront to save files in.
- You can enable logging as an option that you specify when you're creating a distribution.
- One way to analyze your access logs is to use Amazon Athena. Athena is an interactive query service that can help you analyze data for AWS services, including CloudFront.
- AWS recommends that you use the logs to understand the nature of the requests for your content, not as a complete accounting of all requests. CloudFront delivers access logs on a best-effort basis.

Review Topic : CloudFront

Using AWS CloudTrail to Capture Requests Sent to the CloudFront API

- CloudFront is integrated with CloudTrail, an AWS service that captures information about every request that is sent to the CloudFront API by your AWS account, including your IAM users.

- CloudTrail periodically saves log files of these requests to an Amazon S3 bucket that you specify.

- CloudTrail captures information about all requests, whether they were made using the CloudFront console, the CloudFront API, the AWS SDKs, the CloudFront CLI, or another service, for example, AWS CloudFormation.

- You can use information in the CloudTrail log files to determine which requests were made to CloudFront, the source IP address from which each request was made, who made the request, when it was made, and so on.

Note

- To view CloudFront requests in CloudTrail logs, you must update an existing trail to include global services.

CloudFront Reports

CloudFront Cache Statistics Reports

- The CloudFront cache statistics report includes the following information:
  - Total Requests for all HTTP methods
  - Percentage of Viewer Requests by Result Type – Shows hits, misses, and errors
  - Bytes Transferred to Viewers
  - HTTP Status Codes – Shows viewer requests by HTTP status code
  - Percentage of GET Requests that Didn't Finish Downloading

CloudFront Popular Objects Report

- The CloudFront popular objects report lists the 50 most popular objects and statistics about those objects. (no. of requests, hits/misses, hit ratio, no of bytes served, no of incomplete downloads)

Review Topic : CloudFront

CloudFront Reports

CloudFront Top Referrers Report

- The CloudFront top referrers report includes the top 25 referrers, the number of requests from a referrer, and the number of requests from a referrer as a percentage of the total number of requests during the specified period.

CloudFront Usage Reports

- The CloudFront usage reports include the following information:
  - Number of Requests
  - Data Transferred by Protocol
  - Data Transferred by Destination

CloudFront Viewers Reports

- The CloudFront viewers reports include the following information:
- Types of Devices , Browsers, Locations, and Operating Systems that accessed your content.

Amazon CloudFront

Perfect Forward Secrecy Used by ELB and CloudFront

Review Topic : CloudFront

CloudFront Perfect Forward Secrecy

- Several AWS services offer more advanced cipher suites that use the Elliptic Curve Diffie-Hellman Ephemeral (ECDHE) protocol.
- ECDHE allows SSL/TLS clients to provide Perfect Forward Secrecy, which uses session keys that are ephemeral and not stored anywhere.
  - This helps prevent the decoding of captured data by unauthorized third parties, even if the secret long-term key itself is compromised.
- Clients using CloudFront APIs must support Transport Layer Security (TLS) 1.0 or later.
- Clients must also support cipher suites with **perfect forward secrecy (PFS)** such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE).
- Most modern systems such as Java 7 and later support these modes.