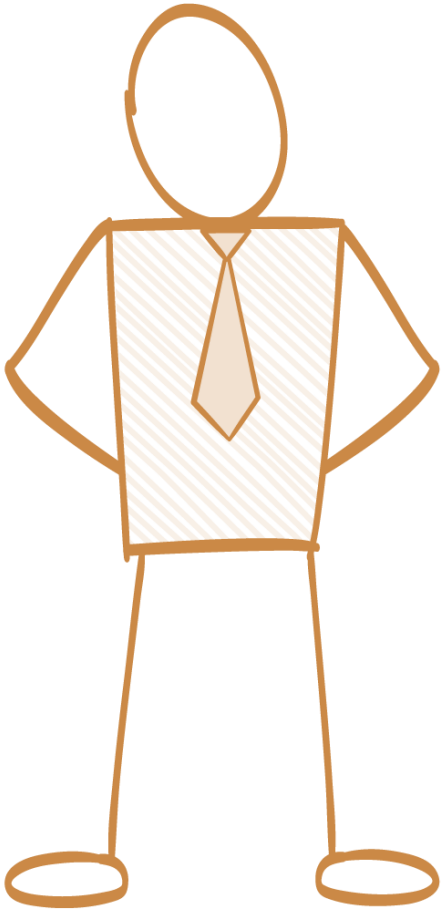


Module 6: Overview of Services for Web Applications

amazon web services

Topics

- AWS products for network content and delivery
  - Amazon Route 53
  - Amazon Elastic Load Balancer
  - Amazon CloudFront
- AWS products for deployment and management
  - Amazon CloudWatch
  - Amazon Elastic Beanstalk
  - AWS CloudFormation

Topics

- AWS products for network content and delivery
  - Amazon Route 53
  - Amazon Elastic Load Balancer
  - Amazon CloudFront
- AWS products for deployment and management
  - Amazon CloudWatch
  - Amazon Elastic Beanstalk
  - AWS CloudFormation

AWS Solutions Architect Associate

DNS

Introduction to Domain Name System

amazon web services

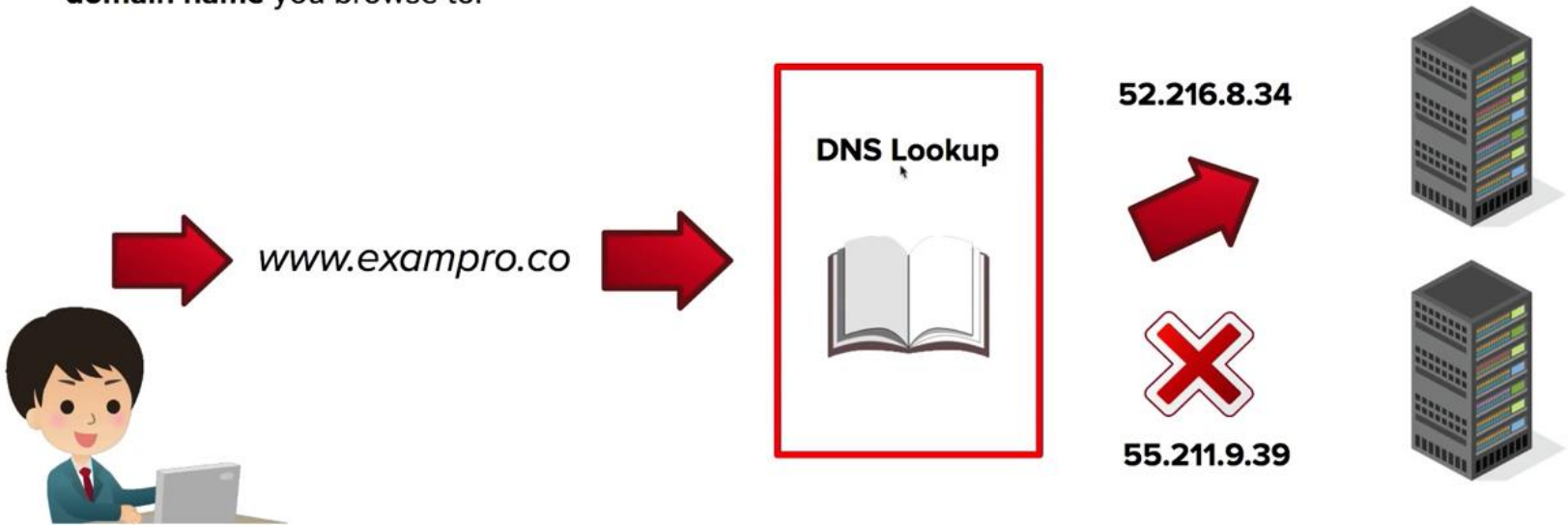
Domain Name System - DNS

The **Phonebook** of the Internet
DNS **translates domain names to IP addresses,**
so browsers can find Internet resources.

Introduction to DNS

Domain Name System (DNS) is the service which handles **converting** a domain name (ie exampro.co) into a routable **Internet Protocol (IP)** address (ie 52.216.8.34)

This is what allows your computer to **find specific servers** on the internet automatically **depending what domain name** you browse to.



AWS Solutions Architect Associate

DNS

Internet Protocol (IP)

amazon web services

Internet Protocol (IP)

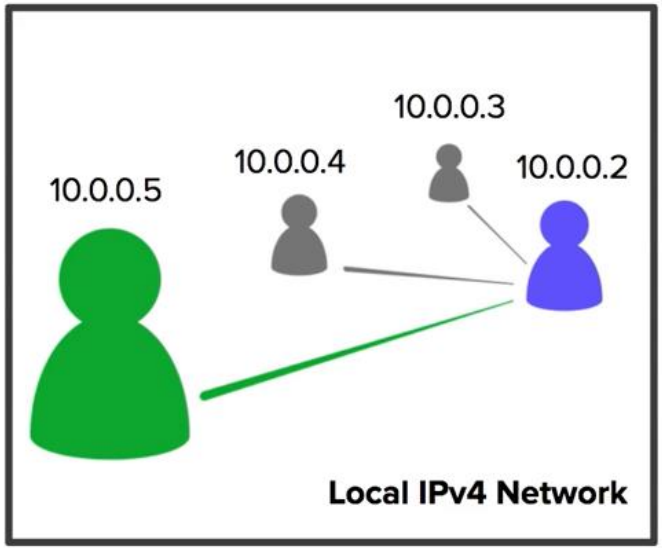
IP Addresses are what uniquely **identifies each computer** on a network, and **allows communication** between them using the Internet Protocol (IP).

IPv4 Internet Protocol Version 4

Example: **52.216.8.34**
Address space is **32-bits** with up to **4,294,967,296** available addresses (we are running out)

IPv6 Internet Protocol Version 6

Example: **2001:0db8:85a3:0000:0000:8a2e:0370:7334**
Address space is **128-bits** with up to **340 undecillion potential addresses (1 + 36 Zeros)** Invented to solve available address limitations of IPv4



AWS Solutions Architect Associate

DNS

Domain Registrars

amazon web services

Domain Registrars

Domain registrars are authorities who **have the ability to assign domain names** under one or more **top-level domains.**

ICANN

Domains get registered through **InterNIC,** which is a service provided by the **Internet Corporation for Assigned Names and Numbers (ICANN),** and enforces the uniqueness of domain names all over the internet.

After registration all domain names can be found publically in a central **WhoIS database.**

Some Popular Domain Registrars You May Know...

HostGator Go Daddy.com DOMAIN.COM aws namecheap

amazon web services

AWS Solutions Architect Associate

DNS

Top-Level Domains

amazon web services

Top-Level Domains

The **last word** within a domain name represents the **top-level** domain name. example.**com**

The **second word** within a domain name is know as the **second-level** domain name. example.**co**.uk

Top-level domain names are controlled by the **Internet Assigned Numbers Authority (IANA)**

Internet Assigned Numbers Authority

All available top level domains are stored in a publically available database at http://www.iana.org/domains/root/db

DOMAIN	TYPE	TLD MANAGER
.aaa	generic	American Automobile Association, Inc.
.aarp	generic	AARP
.abarth	generic	Fiat Chrysler Automobiles N.V.
.abb	generic	ABB Ltd
.abbott	generic	Abbott Laboratories, Inc.
.abbvie	generic	AbbVie Inc.
.abc	generic	Disney Enterprises, Inc.
.able	generic	Able Inc.
.abogado	generic	Minds + Machines Group Limited
.abudhabi	generic	Abu Dhabi Systems and Information Centre
.ac	country-code	Network Information Center (AC Domain Regis Wireless (Ascension Island)
.academy	generic	Binky Moon, LLC

AWS has their own top level domain **.aws** because of course they do.

amazon web services

AWS Solutions Architect Associate

DNS

Start of Authority (SOA)

amazon web services

Start of Authority (SOA)

Every domain **must have an SOA record**. The SOA is a way for the Domain Admins to provide information about the domain eg.: how often it is updated, what is the admin's email address and etc..

A **Zone** file can contain only one SOA Record.

Format:

[authority-domain] [domain-of-zone-admin] [zone-serial-number] [refresh-time] [retry-time] [expire-time] [negative caching TTL]

Example:

ns.example.net. hostmaster.example.com. 1 7200 900 1209600 86400

AWS Example:

ns-415.awsdns-51.com. awsdns-hostmaster.amazon.com. 1 7200 900 1209600 86400

Structure of SOA

NAME	name of the zone
IN	zone class (usually IN for internet)
SOA	abbreviation for Start of Authority
NNAME	Primary master name server for this zone
RNAME	Email of the admin responsible for this zone
SERIAL	Serial number for this zone
REFRESH	seconds after which secondary name servers should query the master for the SOA record, to detect zone changes.
RETRY	seconds after which secondary NS should retry request serial number if unresponsive master
EXPIRE	seconds after which secondary NS should stop answering request for zone if unresponsive master
TTL	Time To Live for purposes of negative caching.

AWS Solutions Architect Associate

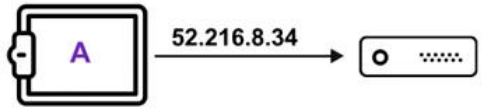
DNS

A Records

amazon web services

Address Records

Address Records (A Records) are one of the fundamental types of DNS records



An A Record allows you to convert the **name of a domain** directly into **an IP address**. They can also be used on the root (naked domain name) itself.

We have testing-domain.com (naked domain name) using an A record to directly to a web-server IP address of 52.216.8.34

```
{
  "ResourceRecordSets": [
    {
      "TTL": 300,
      "Type": "A",
      "Name": "testing-domain.com",
      "ResourceRecords" : [
        { "Value": "52.216.8.34"}
      ]
    }
  ]
}
```

AWS Solutions Architect Associate

DNS

CNAME Records

amazon web services

CNAME Records

Canonical Names (CNAME) are another fundamental DNS record used to **resolve one domain name to another - rather than an IP address**.

The advantage of CNAMES is they are unlikely to change where IP addresses can change over time (if its a dynamic IP address)

We have testing-domain.com (naked domain name) using an A record to redirect our www.testing.domain.com

```
{
  "ResourceRecordSets": [
    {
      "TTL": 300,
      "Type": "CNAME",
      "Name": "testing-domain.com",
      "ResourceRecords" : [
        { "Value": "wwww.testing-domain.com"}
      ]
    }
  ]
}
```

AWS Solutions Architect Associate

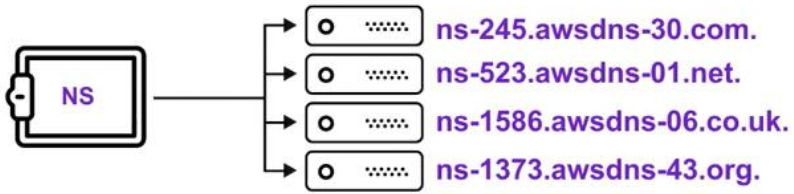
DNS

NS Records

amazon web services

Name Server (NS) Records

Name Server Records (NS) are used by **top-level domain servers** to direct traffic to the DNS server containing the authoritative DNS records. Typically multiple name servers are provided for redundancy.



If you were managing your DNS records with Route53. The **NS records** for your domain name would be pointing at the AWS servers.

These servers are where the DNS records can be found for this domain name

```
{
  "Type": "NS",
  "ResourceRecordSets": [
    {
      "Name": "testing-domain.com",
      "TTL": 172800,
      "ResourceRecords" : [
        { "Value": "ns-245.awsdns-30.com."},
        { "Value": "ns-523.awsdns-01.net."},
        { "Value": "ns-1586.awsdns-06.co.uk."},
        { "Value": "ns-1373.awsdns-43.org."},
      ]
    }
  ]
}
```

amazon web services

AWS Solutions Architect Associate

DNS

Time To Live (TTL)

amazon web services

Time to Live (TTL)

Time-to-live (TTL) is the **length of time that a DNS record gets cached** on the resolving server or the users own local machine.

The lower the TTL - the faster that changes to DNS records will propagate across the internet.

TTL is always measured in seconds under IPv4.

AWS Solutions Architect Associate

amazon web services

DNS *CheatSheet*

- **Domain Name System (DNS)** - Internet service that converts domain names into routable IP addresses
- **IPv4** - Internet Protocol Version 4 - 32 bit address space (**limited** number of addresses)
- IPv4 eg. **52.216.8.34**
- **IPv6** - Internet Protocol Version 6 - 128 bit address space (**unlimited** number of addresses)
- IPv6 eg. **2001:0db8:85a3:0000:0000:8a2e:0370:7334**
- **Top-Level Domain** example**.com** last part of the domain
- **Second-Level Domain** example**.CO**.UK **second last part of the domain**
- **Domain Registrar** 3rd party company who you register domains through
- **Name Server** The server(s) which contain the DNS records for a domain
- **Start of Authority (SOA)** Contains information about the DNS zone and associated DNS records
- **A Record** DNS record which directly converts a domain name into an IP address
- **CNAME Record** DNS record which lets you convert a domain name into another domain name
- **Time to Live (TTL)** The time that a DNS record will be cached for (lower time means changes propagate faster)

AWS Solutions Architect Associate

Route53

Introduction to Route53

amazon web services

Route53

Highly available and scalable cloud Domain Name System (DNS).
Register and manage domains, create DNS routing rules eg. failovers.

amazon web services

Route53 – Introduction

Route53 is a DNS is a Domain Name Service ***think*** Godaddy or NameCheap but with more synergies with AWS Services.

You can:

- register and manage domains
- create various records sets on a domain
- Implement complex traffic flows eg. Blue/green deploy, failovers
- Continuously monitor records via health checks
- resolve VPC's outside of AWS

AWS Solutions Architect Associate

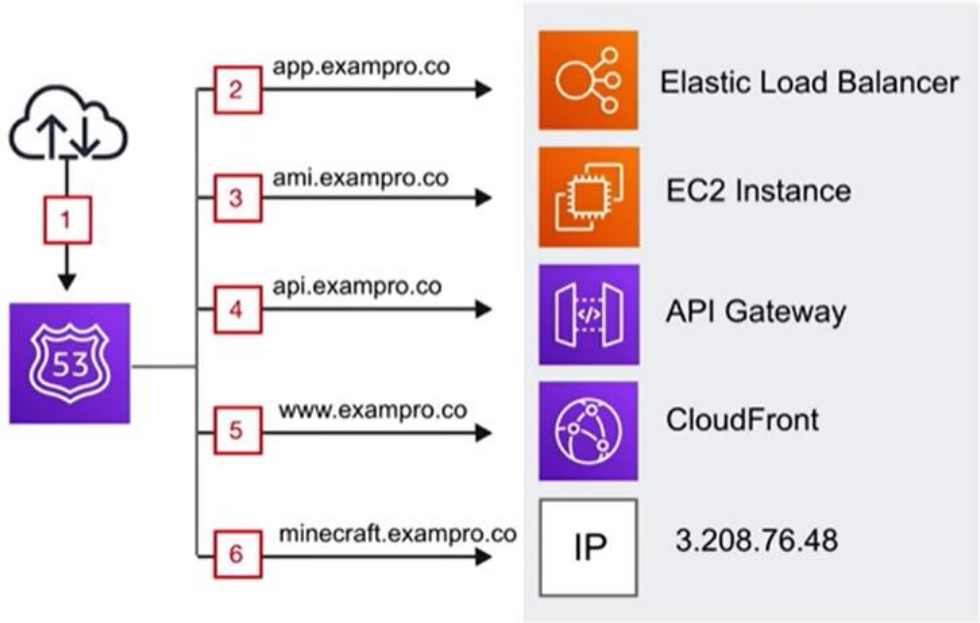
Route53

Route53 Use Case

Route53 - Use Case

Use Route53 to get your custom domains to point to your AWS Resources

1. Incoming internet traffic
2. Route traffic to our web-app backed by ALB
3. Route traffic to an instance we use to tweak our AMI
4. Route traffic to API gateway which powers our API
5. Route traffic to CloudFront which serves our S3 static hosted website
6. Route traffic to an Elastic IP (EIP) which is a static IP that hosts our company Minecraft server



AWS Solutions Architect Associate

Route53

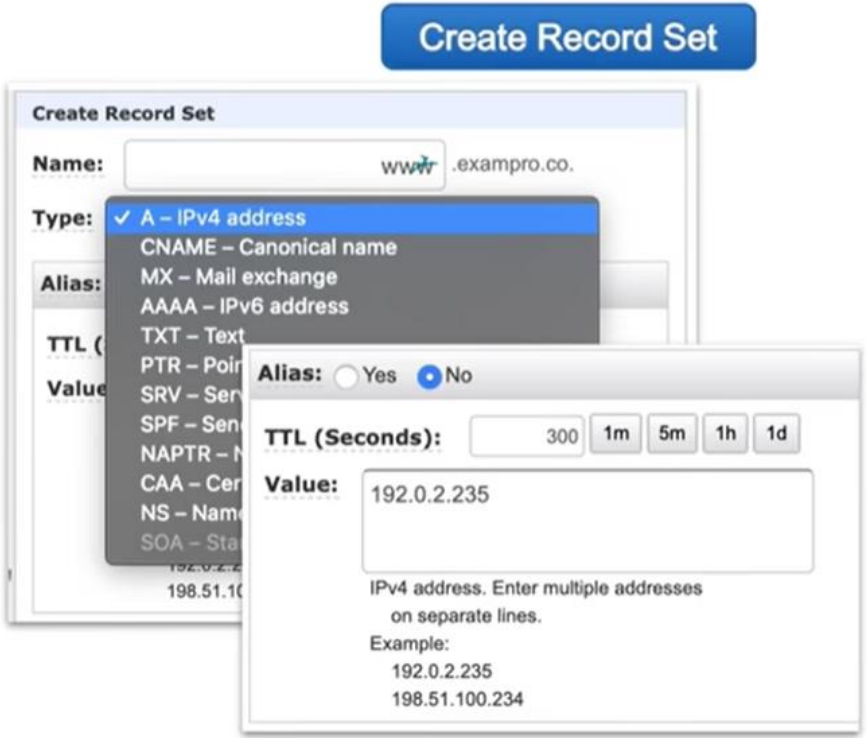
Record Sets

amazon web services

Route53 - Record Sets

We create record sets which allows us to point our naked domain (exampro.co) and subdomains via Domain records.

For example we can send our www subdomain using an A record to point a specific IP address.

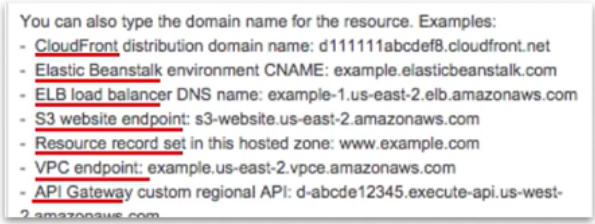


Route53 - Alias Record

AWS has their own special Alias Record which extends DNS functionality. It will route traffic to specific AWS resources.

Alias records are smart where they can detect the change of an IP address and continuously keep that endpoint pointed to the correct resource.

In most cases you want to be using **Alias** when routing traffic to AWS resources



amazon web services

AWS Solutions Architect Associate

Route53

Routing Policies

amazon web services

Route53 – Routing Policies

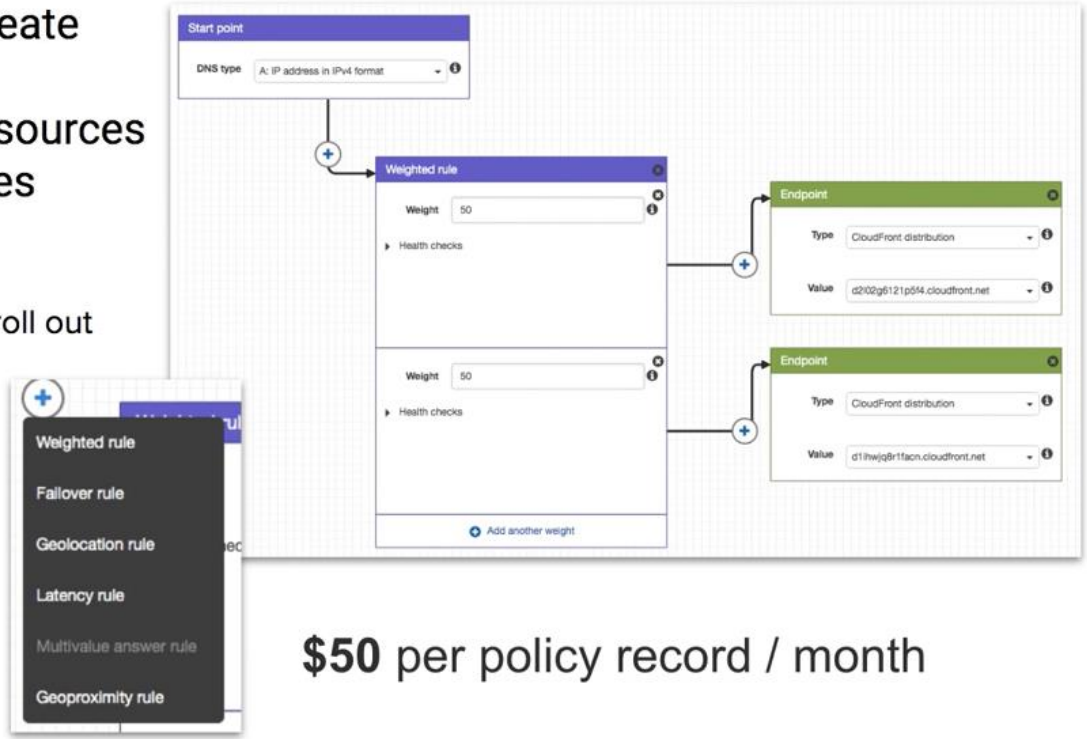
There are **7 different types** of Routing Policies available inside Route53

- **Simple Routing** default routing policy, multiple addresses result in random selection
- **Weighted Routing** route traffic based on weighted values to split traffic
- **Latency-Based Routing** route traffic to region resource with lowest latency
- **Failover Routing** route traffic if primary endpoint is unhealthy to secondary endpoint
- **Geolocation Routing** route traffic based on the location of your users
- **Geo-proximity Routing** route traffic based on the location of your resources and, optionally, shift traffic from resources in one location to resources in another
- **Multi-value Answer Routing** respond to DNS queries with up to eight healthy records selected at random.

Route53 – Traffic Flow

A visual editor lets you create sophisticated routing configurations for your resources using existing routing types

Supports **versioning** so you can roll out or roll back updates.



$50 per policy record / month

amazon web services

AWS Solutions Architect Associate

Route53

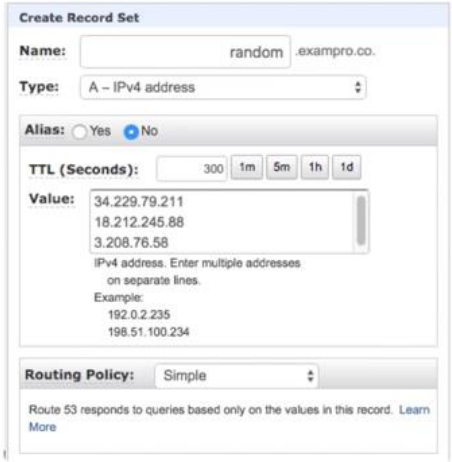
Simple Routing Policies

amazon web services

Route53 – Simple Routing Policies

Simple Routing Policies are the most basic routing policies in Route53 **Default Policy**

- You have 1 record and provide multiple IP addresses
- When multiple values are specified for a record, Route53 will return all values back to the user in a **random order**



For example if you had a record for 'www.exampro.co' with 3 different IP address values, users would be directly **randomly to 1 of them** when visiting the domain.

AWS Solutions Architect Associate

Route53

Weighted Routing Policies

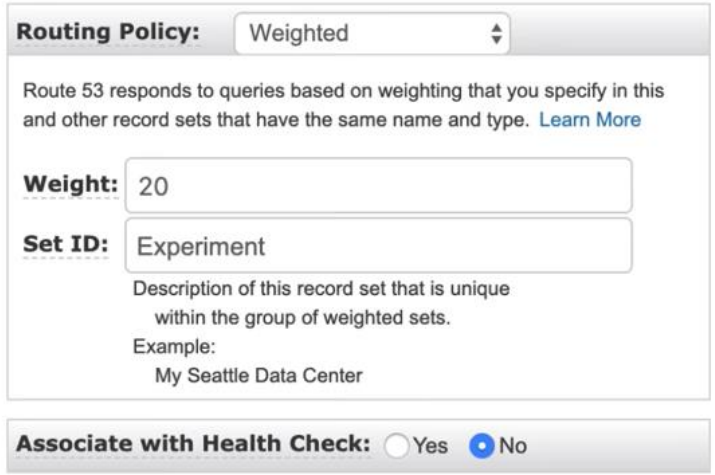
amazon web services

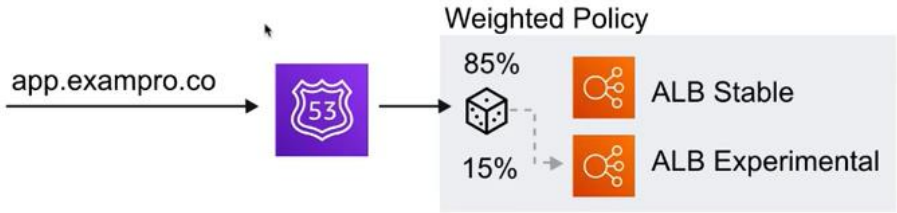
Route53 - Weighted Routing Policies

Weighted Routing Policies let you split up traffic based on different 'weights' assigned.

This allows you to send a certain percentage of overall traffic to one server, and have any other traffic apart from that directed to a completely different server.

For example if you had an ALB running experimental features you could test against a small amount traffic at random to minimize the impact of affect





AWS Solutions Architect Associate

Route53

Latency Based Routing

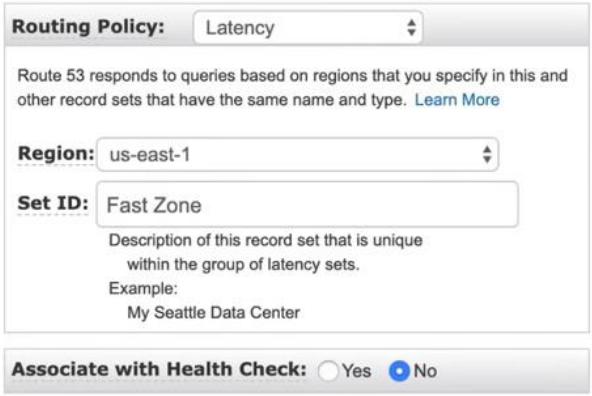
amazon web services

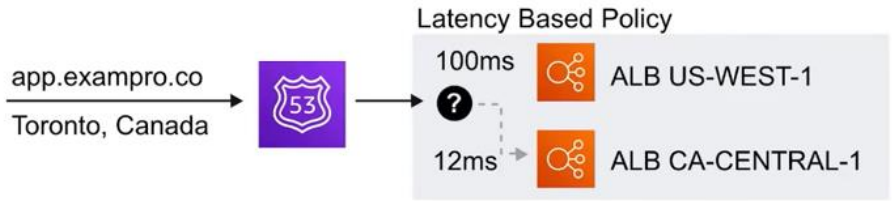
Route53 – Latency Based Routing Policies

Latency Based Routing allows you to direct traffic based on the lowest network latency possible for your end-user **based on region**.

Requires a latency resource record to be set for the EC2 or ELB resource that hosts your application in each region.

For example, You have two copies of your web-app backed by ALB. One in California, US and another in Montreal, Canada. An request comes in from Toronto, it will be routed to Montreal since it will have lower latency





AWS Solutions Architect Associate

Route53

Failover Routing Policies

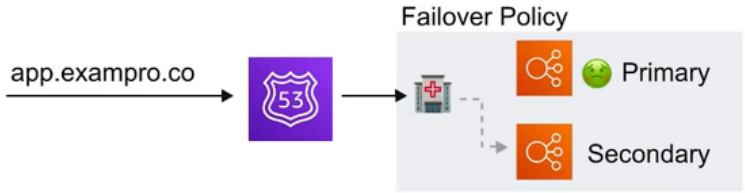
amazon web services

Route53 – Failover Routing Policies

Failover Routing Policies allow you to create active/passive setups in situations where you want a primary site in one location, and a secondary data recovery site in another.

Route53 automatically monitors health-checks from your primary site to determine the health of end-points. If an end-point is determined to be in a failed state, all traffic is automatically directed to the secondary location.

For example, we have a primary and secondary web-app backed by ALB. Route53 determines our primary is unhealthy and fails over to secondary ALB.



AWS Solutions Architect Associate

Route53

Geolocation Routing Policies

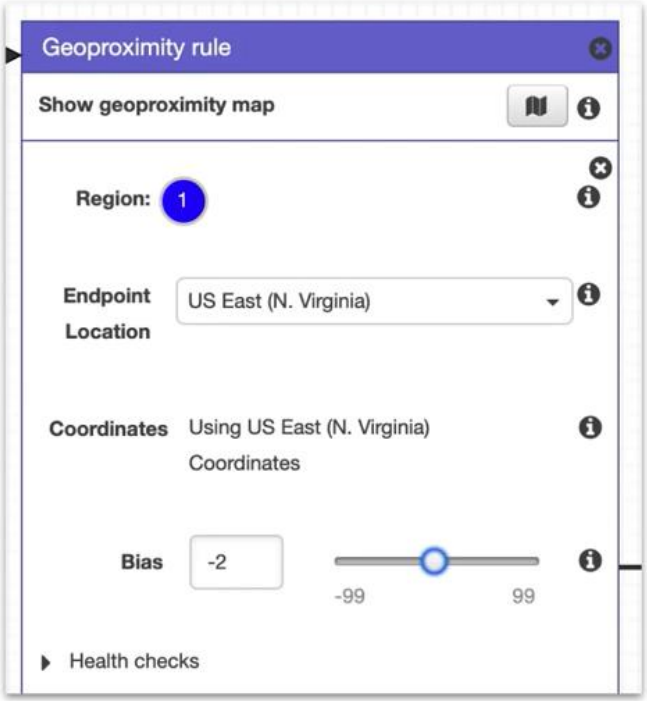
amazon web services

Route53 – Geoproximity Routing Policies

Geoproximity Routing Policies allow you to direct traffic based on the geographic location of your users, and your AWS resources.

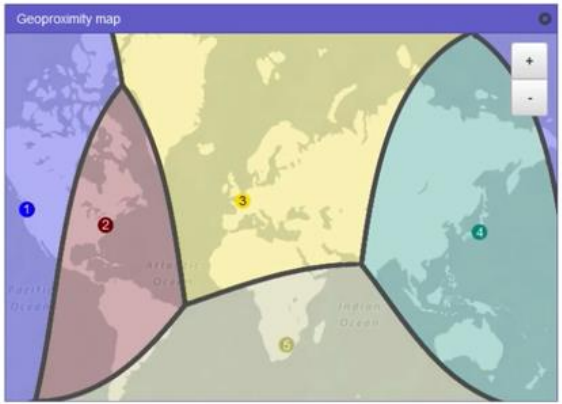
You can route more or less traffic to a specific resource by specifying a 'Bias' value.

Bias values expand or shrink the size of the geographic region from which traffic is routed to. **You must use Route53 Traffic Flow** in order to use geoproximity routing policies

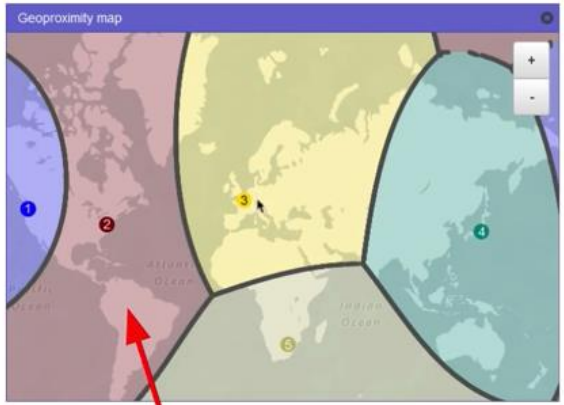


Route53 – Geoproximity Routing Policies

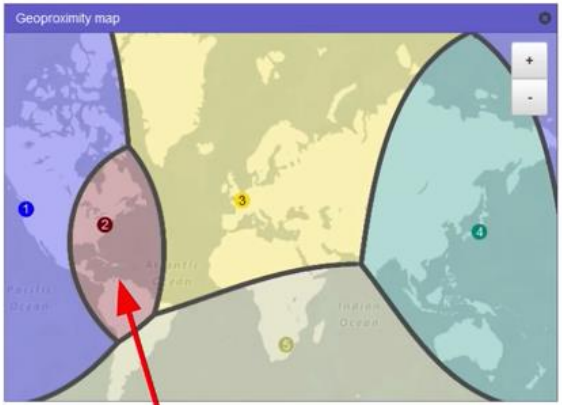
How Bias Works for Geoproximity



Standard Bias



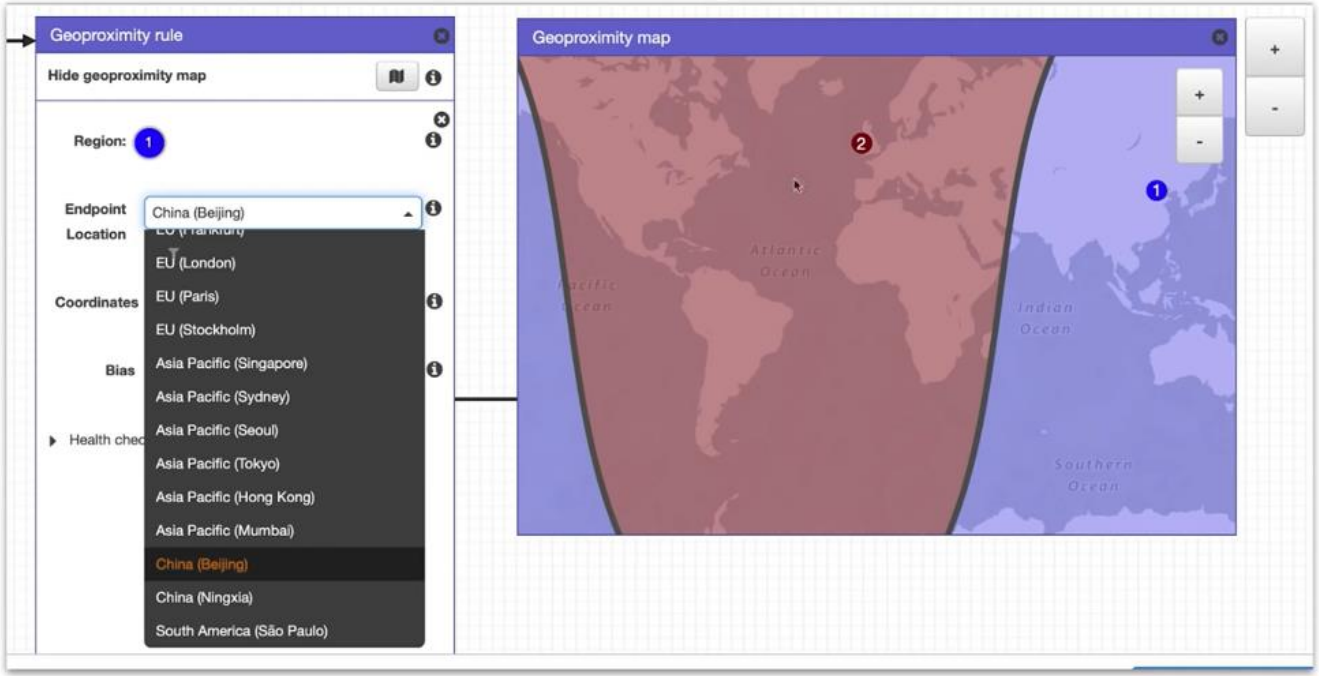
+25 for the US East



-25 for the US East

Route53 – Geoproximity Routing Policies

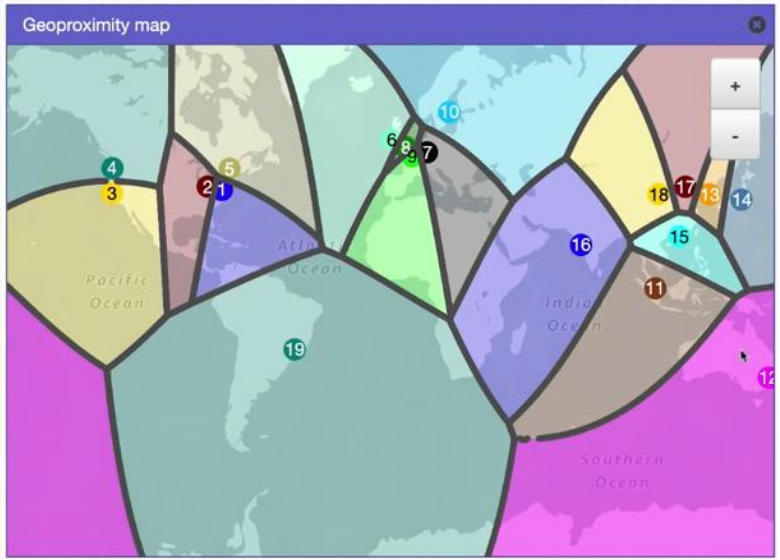
In the Route53 Traffic Flow you can select any regions and visualize the bias



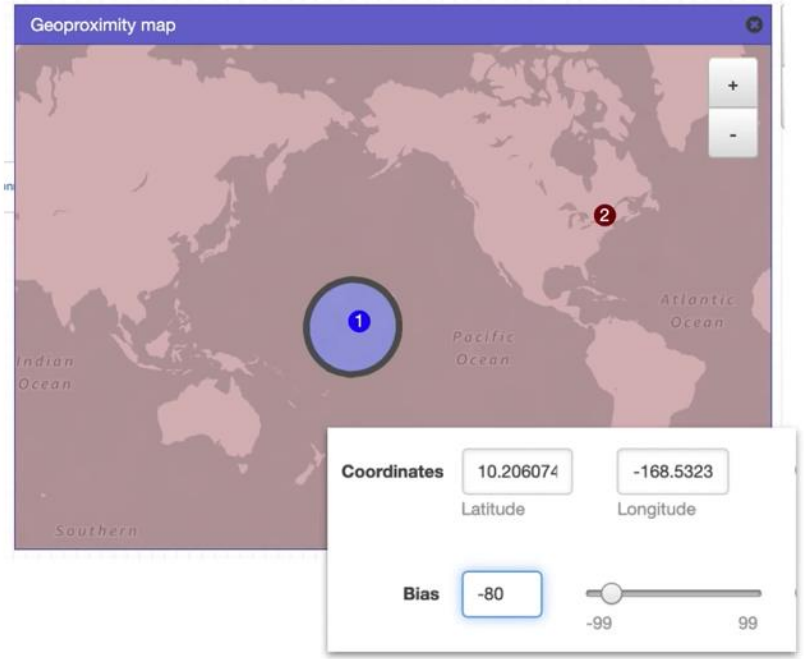
amazon web services

Route53 – Geoproximity Routing Policies

All Current Regions Selected



Can provide **custom coordinates** over region



AWS Solutions Architect Associate

Route53

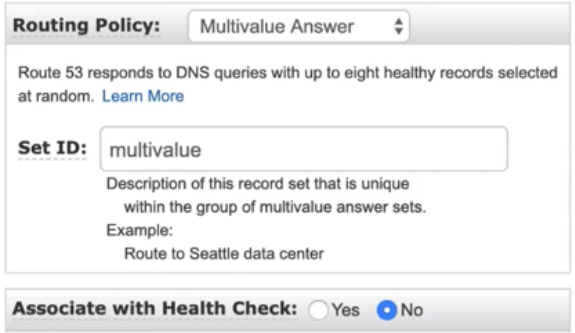
Multi-Value Answer Policies

amazon web services

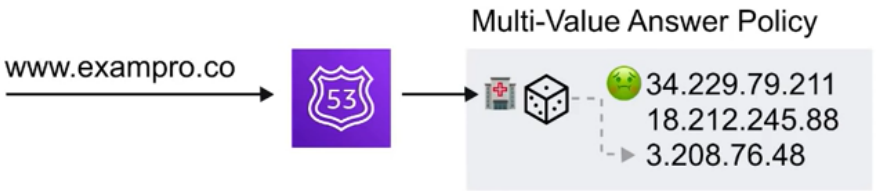
Route53 - Multi-Value Answer Policies

Multi-Value Answer Policies let you configure Route53 to return multiple values such as IP addresses for your web-servers, in response to DNS queries.

Multiple values can be specified for almost any record. Route53 automatically performs health-checks on resources and only returns values of ones deemed healthy.



Similar to Simple Routing, however with an added health check for your record set resources.



AWS Solutions Architect Associate

Route53

Health Checks

amazon web services

Route53 - Health Checks

- Checks health every **30s** by default. Can be reduce to every **10s**
- A health check can **initial a failover** if status is returned unhealthy
- A CloudWatch Alarm can be created to alert you of status unhealthy
- A health check can **monitor other health checks** to create a chain of reactions.

can create up to **50 health checks** for AWS endpoints that are within or linked to the same AWS account.

	AWS Endpoints	Non-AWS Endpoints
Basic Health Checks	$0.50* per health check / month	$0.75 per health check / month
Optional health check features: • HTTPS • String Matching • Fast Interval • Latency Measurement	$1.00 / month per optional feature	$2.00 / month per optional feature

AWS Solutions Architect Associate

Route53

Route53 Resolver

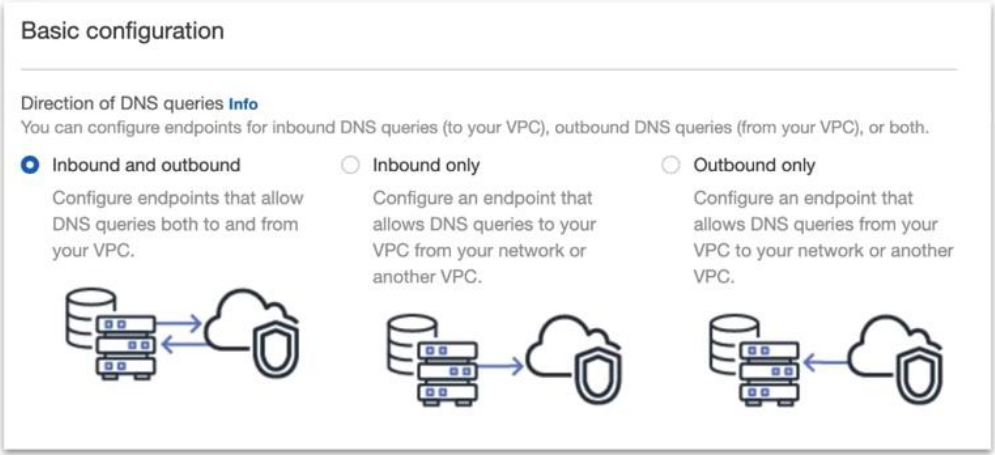
amazon web services

Route53 – Resolver

Formally known as **.2 resolver**

A regional service that lets you route DNS queries between your VPCs and your network

DNS Resolution for **Hybrid Environments** (On-Premise and Cloud)



AWS Solutions Architect Associate

Route53

Route53 Cheat Sheet

amazon web services

Route53 CheatSheet

- Route53 is a DNS provider, register and manage domains, create record sets. Think Godaddy or NameCheap
- Simple Routing - Default routing policy, multiple addresses result in a random endpoint selection
- Weighted Routing - Split up traffic based on different 'weights' assigned (percentages)
- Latency-Based Routing - Directs traffic based on region, for lowest possible latency for users.
- Failover Routing - Primary site in one location, secondary data recovery site in another. (change on health check)
- Geolocation Routing - Route traffic based on the geographic location of a requests origin.
- Geo-proximity Routing - Route traffic based on geographic location using 'Bias' values (needs Route53 Traffic Flow)
- Multi-value Answer Routing - Return multiple values in response to DNS queries. (using health checks)
- Traffic Flow - visual editor, for chaining routing policies, can version policy records for easy rollback
- AWS Alias Record - AWS' smart DNS record, detects changed IPs for AWS resources and adjusts automatically.
- Route53 Resolver - Lets you regionally route DNS queries between your VPCs and your network **Hybrid Environments**
- Health checks can be created to monitor and automatically over endpoints. You can have health checks monitor other health checks

Network and Content Delivery

- Amazon Route 53 
- Amazon Elastic Load Balancer 
- Amazon CloudFront 

AWS Solutions Architect Associate

Elastic Load Balancer

Elastic Load Balancer Introduction

amazon web services

Elastic Load Balancer (ELB)

Distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, IP addresses, and Lambda functions.

Introduction to Elastic Load Balancer (ELB)

Load Balancers can be physical hardware or virtual software that accepts incoming traffic, and then distributes the traffic to multiple targets. The can **balance** the load via different rules. These rules vary based on types of load balancers.

Elastic Load Balancer (ELB) is the AWS solution for load balancing traffic, and there are 3 types available:

1. Application Load Balancer ALB (HTTP/HTTPS)
2. Network Load Balancer NLB (TCP/UDP)
3. Classic Load Balancer CLB (Legacy)

AWS Solutions Architect Associate

Elastic Load Balancer

ELB Rules of Traffic

amazon web services

ELB – The Rules of Traffic

Listeners

Incoming traffic is evaluated against listeners. Listeners evaluate any traffic that is matches the Listener's port. For Classic Load Balancer, EC2 instances are directly registered to the Load Balancer.

Rules (Not avaliable for Classic Load Balancer)

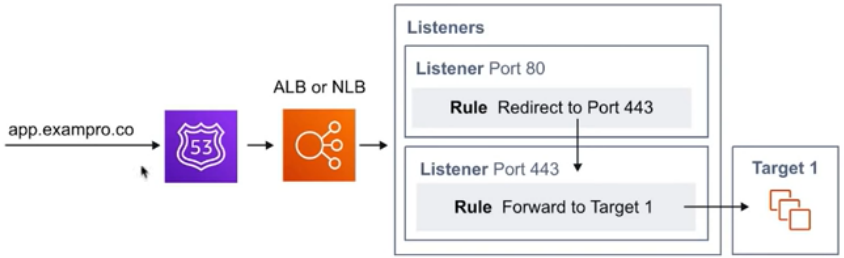
Listeners will then invoke rules to decide what to do with the traffic. Generally the next step is to forward traffic to a Target Group

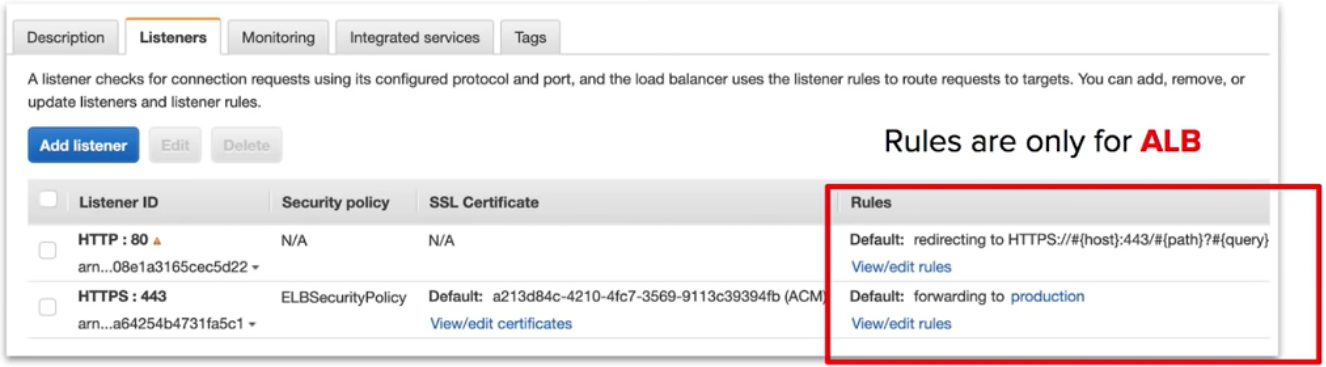
Target Groups (Not avaliable for Classic Load Balancer)

EC2 instances are registered as targets to a Target Group

ELB - The Rules of Traffic

For Application Load Balancer (ALB) or Network Load Balancer (NLB) traffic is sent to the Listeners. When the port matches it then checks the rules what do to. The rules will forward the traffic to a Target Group. The target group will evenly distribute the traffic to instances registered to that target group.





amazon web services

ELB - The Rules of Traffic

For Classic Load Balancer (CLB) traffic is sent to the Listeners. When the port matches it then it forwards the traffic to any EC2 instances that are registered to the Classic Load Balancer. CLB does not allow you to apply rules to listeners

AWS Solutions Architect Associate

Elastic Load Balancer

Application Load Balancer (ALB)

amazon web services

Application Load Balancer (ALB)

Application Load Balancers are designed to balance **HTTP** and **HTTPS** traffic.

They **operate at Layer 7 (of the OSI Model)**.

ALB has a feature called **Request Routing** which allows you to add routing rules to your listeners based on the HTTP protocol.

Web Application Firewall (WAF) can be attached to ALB.

Great for Web Applications

OSI Layers

Layer 7	**Application**
Layer 6	**Presentation**
Layer 5	**Session**
Layer 4	**Transport**
Layer 3	**Network**
Layer 2	**Data Link**
Layer 1	**Physical**

AWS Solutions Architect Associate

Elastic Load Balancer

Network Load Balancer (NLB)

amazon web services

Network Load Balancer (NLB)

Network Load Balancers are designed to balance **TCP/UDP**.

They **operate at Layer 4 (of the OSI Model)**

Can handle **millions of requests per second** while still maintaining extremely low latency.

Can preform Cross-Zone Load Balancing

Great for Multiplayer Video Games or When network performance is critical

OSI Layers

Layer 7	**Application**
Layer 6	**Presentation**
Layer 5	**Session**
Layer 4	**Transport**
Layer 3	**Network**
Layer 2	**Data Link**
Layer 1	**Physical**

AWS Solutions Architect Associate

Elastic Load Balancer

Classic Load Balancer (CLB)

amazon web services

Classic Load Balancer (CLB)

It was AWS first load balancer **(legacy)**

Can balance **HTTP, HTTPS** or **TCP** traffic (not at the same time)

It can use **Layer 7-specific features (OSI Model)** such as **sticky sessions**.

It can also use **strict Layer 4 (OSI Model)** balancing for purely TCP applications.

Can preform Cross-Zone Load Balancing

It will respond with a **504 error (timeout)** if the underlying application is not responding. (**at the web-server or database level**)

Not recommended for use, instead use NLB or ALB

OSI Layers

Layer 7	**Application**
Layer 6	**Presentation**
Layer 5	**Session**
Layer 4	**Transport**
Layer 3	**Network**
Layer 2	**Data Link**
Layer 1	**Physical**

AWS Solutions Architect Associate

Elastic Load Balancer

Sticky Sessions

amazon web services

ELB – Sticky Sessions

Sticky Sessions is an advanced load balancing method that allows you to **bind a user's session to a specific EC2 instance.**

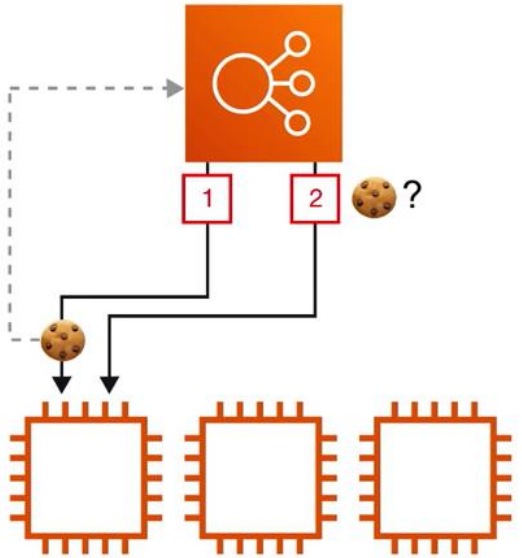
Ensures all **requests** from that session are **sent to the same instance.**

Typically **utilized** with a **Classic Load Balancer**

Can be enabled for ALB though can only be set on a Target Group not individual EC2 instances.

Cookies are used to remember which EC2 instance.

Useful when specific **information is only stored locally on a single instance**



AWS Solutions Architect Associate

Elastic Load Balancer

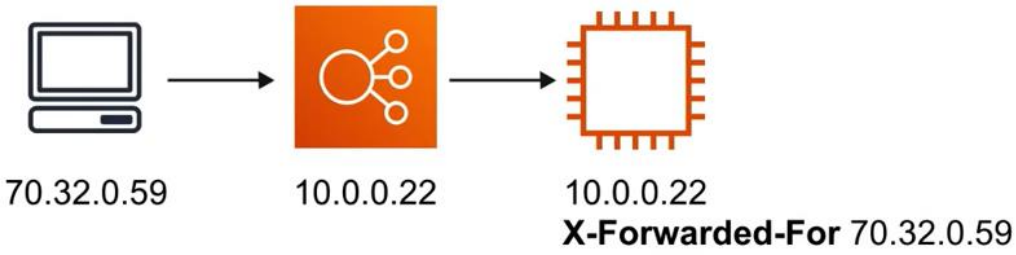
X-Forwarded-For Header

amazon web services

X-Forwarded-For (XFF) Header

If you **need the IPv4 address** of a user, check the **X-Forwarded-For** header

The **X-Forwarded-For (XFF)** header is a command method for identifying the **originating IP address** of a client connecting to a web server through an HTTP proxy or a load balancer.



AWS Solutions Architect Associate

Elastic Load Balancer

ELB Health Checks

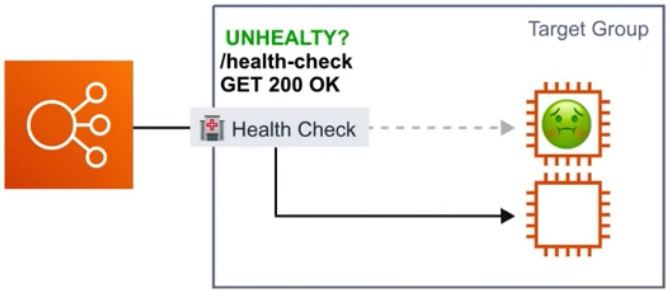
amazon web services

ELB – Health Checks

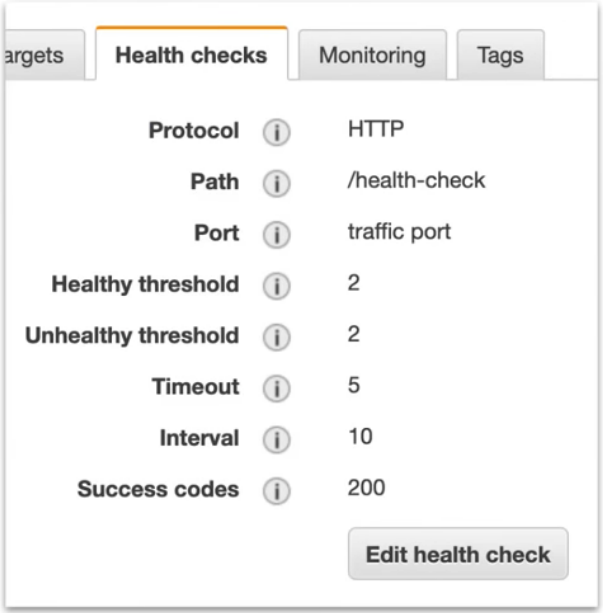
Instances that are monitored by the Elastic Load Balancer (ELB) **report back** Health Checks as **InService, or OutofService**

Health Checks communicate directly with the instance to determine its state.

ELB **does not terminate (kill) unhealthy instance**. It will just redirect traffic to healthy instances



For ALB and NLB the Heath checks are found on the **Target Group**



AWS Solutions Architect Associate

Elastic Load Balancer

Cross-Zone Load Balancing

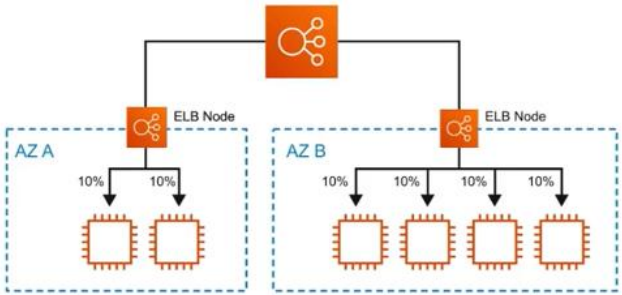
amazon web services

ELB – Cross-Zone Load Balancing

Only for **Classic** and **Network** Load Balancer

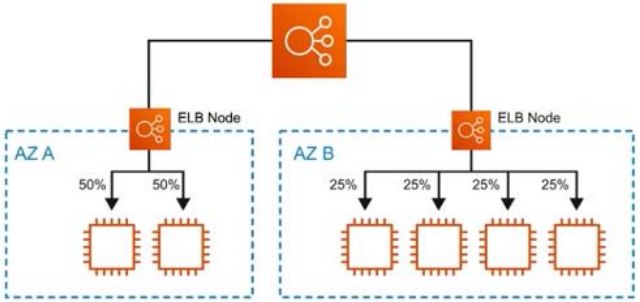
Cross-Zone Load Balancing Enabled

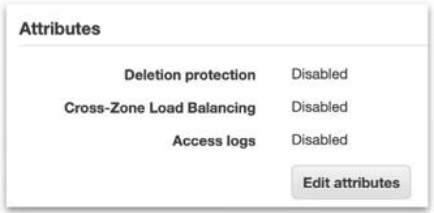
requests are distributed evenly across the instances **in all enabled** Availability Zones.

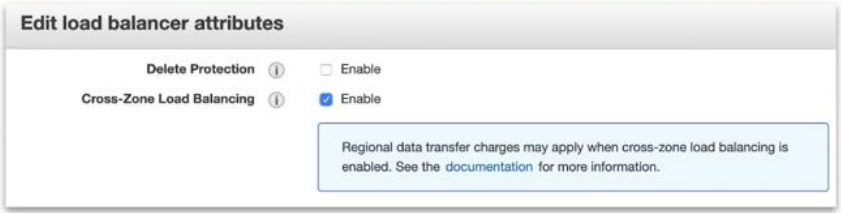


Cross-Zone Load Balancing Disabled

requests are distributed evenly across the instances **in only** its Availability Zone.







amazon web services

AWS Solutions Architect Associate

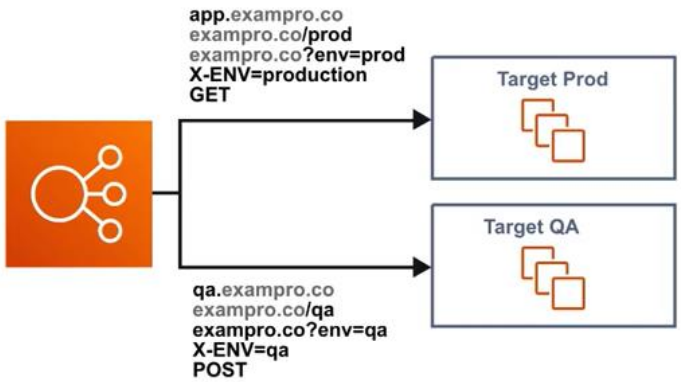
Elastic Load Balancer

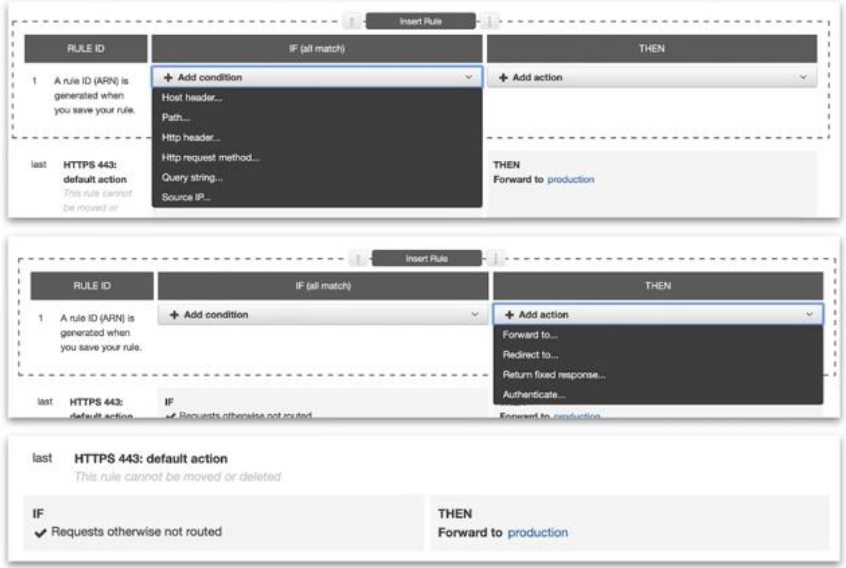
Request Routing

ALB - Request Routing

Apply rules to incoming request and then **forward** or **redirect** traffic.

- ✓ Host header
- ✓ Source IP
- ✓ Path
- ✓ Http header
- ✓ Http header method
- ✓ Query string





amazon web services

AWS Solutions Architect Associate

Elastic Load Balancer

ELB Cheat Sheet

amazon web services

ELB CheatSheet

- There are three Elastic Load Balancers: **Network**, **Application** and **Classic** Load Balancer
- A Elastic Load Balancer must have **at least two** Availability Zones.
- Elastic Load Balancers **cannot go cross-region**. You must create one per region.
- ALB has **Listeners**, **Rules** and **Target Groups** to route traffic
- NLB use **Listeners** and **Target Groups** to route traffic
- CLB use **Listeners** and EC2 instances are **directly registered** as targets to CLB
- Application Load Balancer is for HTTP(S) traffic and the name implies it good for Web Applications
- Network Load Balancer is for TCP/UDP is good for high network throughput eg. Video Games
- Classic Load Balancer is legacy and its recommended to use ALB or NLB
- Use X-Forwarded-For (XFF) to get original IP of incoming traffic passing through ELB
- You can attach Web Application Firewall (WAF) to ALB but not to NLB or CLB
- You can attach Amazon Certification Manager SSL to any of the Elastic Load Balancers for SSL
- ALB has advanced Request Routing rules where you can route based on subdomain header, path and other HTTP(S) information
- Sticky Sessions can be enable for CLB or ALB and sessions are remembered via Cookie

Network and Content Delivery

- Amazon Route 53 
- Amazon Elastic Load Balancer 
- Amazon CloudFront 

AWS Solutions Architect Associate

CloudFront

CloudFront Introduction

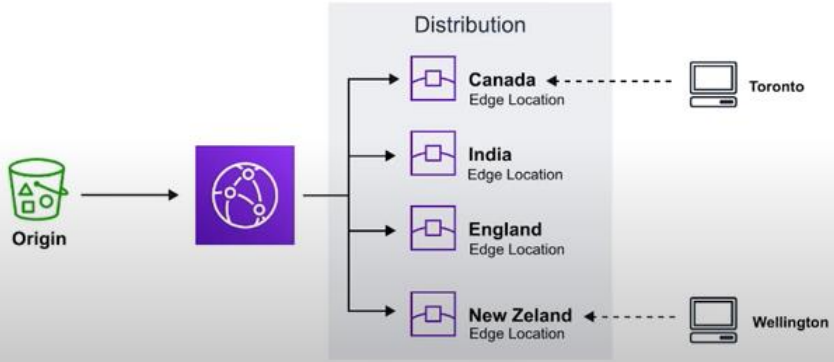
CloudFront

Content Distribution Network (CDN)
Creates cached copies of your website at various Edge Locations around the world

Introduction to CloudFront

Content Delivery Network (CDN)

A CDN is a distributed network of servers which delivers web pages and content to users based on their **geographical location**, the **origin of the webpage**, and a **content delivery server**.



Can be used to **deliver an entire website** including static, dynamic and streaming

Requests for content are served from the nearest Edge Location for the best possible performance.

AWS Solutions Architect Associate

CloudFront

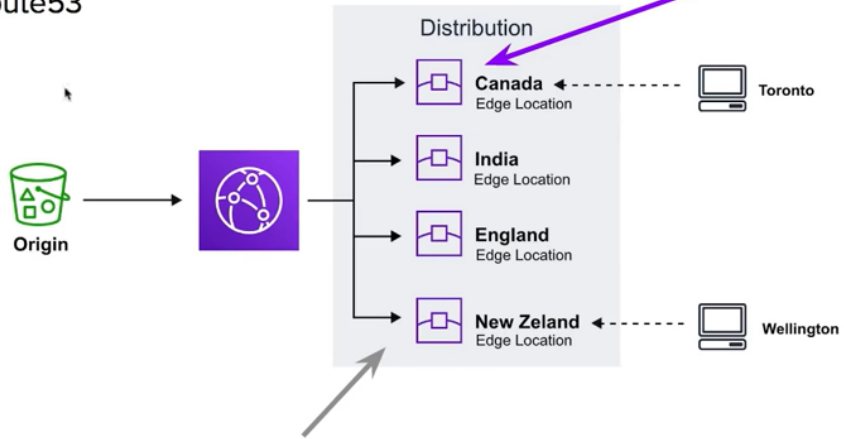
CloudFront Core Components

amazon web services

CloudFront – Core Components

Origin The location where all of original files are located. For example an S3 Bucket, EC2 Instance, ELB, or Route53

Edge Location The location where web content will be cached. This is different than an AWS Region or AZ



Distribution A collection of Edge locations which defines how cached content should behave

AWS Solutions Architect Associate

CloudFront

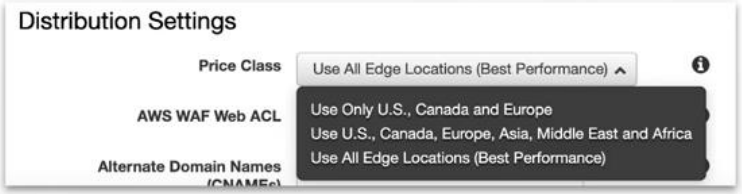
CloudFront Distributions

amazon web services

CloudFront – Distributions

A Distribution is a collection of Edge Locations.
You specific the Origin eg. S3, EC2, ELB, Route53

It replicates copies based on your **Price Class**



There are 2 types ✌ of Distributions

1. Web (for websites)
2. RTMP (for streaming media)

Behaviours

Redirect to HTTPs, Restrict HTTP Methods, Restrict Viewer Access, Set TTLs

Invalidations

You can manually invalidate cache on specific files via Invalidations

Error Pages

You can serve up custom error pages .eg 404

Restrictions

You can use **Geo Restriction** to blacklist or whitelist specific countries

AWS Solutions Architect Associate

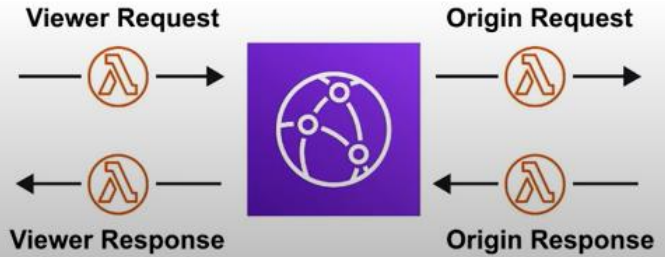
amazon web services

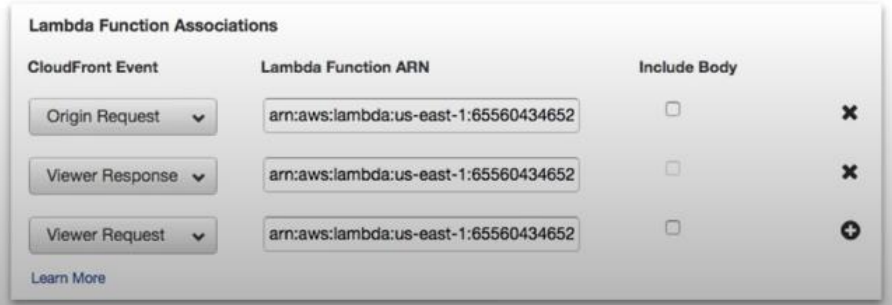
CloudFront – Lamdba@Edge

We use Lambda@Edge functions to **override the behaviour** of **request and responses**

The 4 Available Lambda@Edge Functions

1. **Viewer request** When CloudFront receives a request from a viewer
2. **Origin request** Before CloudFront forwards a request to the origin
3. **Origin response** When CloudFront receives a response from the origin
4. **Viewer response** Before CloudFront returns the response to the viewer





amazon web services

AWS Solutions Architect Associate

CloudFront

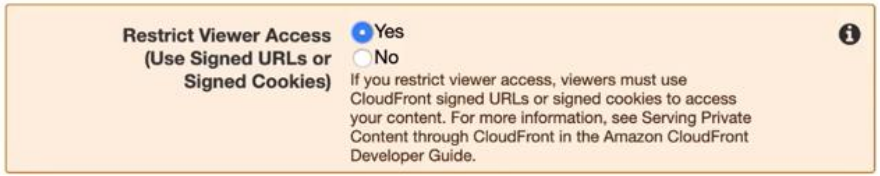
CloudFront Protection

CloudFront - Protection

By Default a Distribution **allows everyone to have access.**

Original Identity Access (OAI)

A virtual user identity that will be used to give your CloudFront Distribution permission to fetch a private object



In order to use Signed URLs or Signed Cookies you need to have an **OAI**

Signed URLs (Not the same thing as S3 Presigned URL)

A url with provides temporary access to cached objects

Signed Cookies

A cookie which is passed along with the request to CloudFront. The advantage of using a Cookie is you want to provide access to multiple restricted files. eg. Video streaming

AWS Solutions Architect Associate

CloudFront

CloudFront Cheat Sheet

amazon web services

CloudFront CheatSheet

- CloudFront is a CDN (Content Distribution Network). It makes website load fast by serving cached content that is nearby
- CloudFront distributes cached copy at **Edge Locations**
- Edge Locations aren't just not read-only, you can write to them eg. PUT objects
- **TTL** (Time to live) defines how how long until the cache expires (refreshes cache)
- When you invalidate your cache, you are forcing it to immediately expire (refreshes cached data)
- Refreshing the cache **costs money because of transfer costs** to update Edge Locations
- **Origin** is the address of where the original copies of your files reside eg. S3, EC2, ELB, Route53
- **Distribution** defines a collection of Edge Locations and behaviour on how it should handle your cached content
- **Distributions** has 2 Types: **Web Distribution** (static website content) **RTMP** (streaming media)
- **Origin Identity Access (OAI)** is used access private S3 buckets
- Access to cached content can be protected via **Signed Urls** or **Signed Cookies**
- **Lambda@Edge** allows you to pass each request through a Lambda to change the behaviour of the response.

Topics

- AWS products for network content and delivery
  - Amazon Route 53
  - Amazon Elastic Load Balancer
  - Amazon CloudFront
- AWS products for deployment and management
  - Amazon CloudWatch
  - Amazon Elastic Beanstalk
  - AWS CloudFormation

AWS Products for Deployment and Management

- Amazon CloudWatch 
- Amazon Elastic Beanstalk 
- AWS CloudFormation 

AWS Solutions Architect Associate

CloudWatch

CloudWatch Introduction

amazon web services

CloudWatch

A collection of monitoring services for logging, reacting and visualizing log data.

amazon web services

Introduction to CloudWatch

AWS CloudWatch is a **monitoring solution** for your AWS resources

CloudWatch is a **collection of monitoring tools** as follows:

CloudWatch **Logs**	any custom log data, Memory Usage, Rails Logs, Nginx Logs
CloudWatch **Metrics**	metrics that are based off of logs eg. Memory Usage
CloudWatch **Events**	trigger an event based on a condition eg. ever hour take snapshot of server
CloudWatch **Alarms**	triggers notifications based on metrics which breach a defined threshold
CloudWatch **Dashboards**	create visualizations based on metrics

AWS Solutions Architect Associate

CloudWatch

CloudWatch Logs

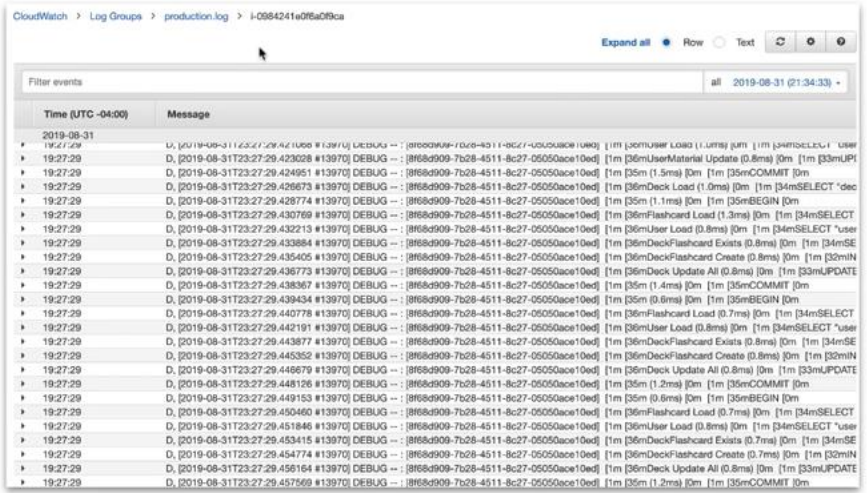
CloudWatch Logs

CloudWatch Logs is used to monitor, store, and access your log files

A **Log Group** is a collection of logs. Log files must belong to a log group

A Log in a Log Group is called a **Log Stream**

By default, **logs are kept indefinitely and never expire**



Most AWS Services are integrated with CloudWatch Logs. Logging of services sometimes needs to be turned on or requires the IAM Permissions to write to CloudWatch Logs

amazon web services

AWS Solutions Architect Associate

CloudWatch

CloudWatch Metrics

amazon web services

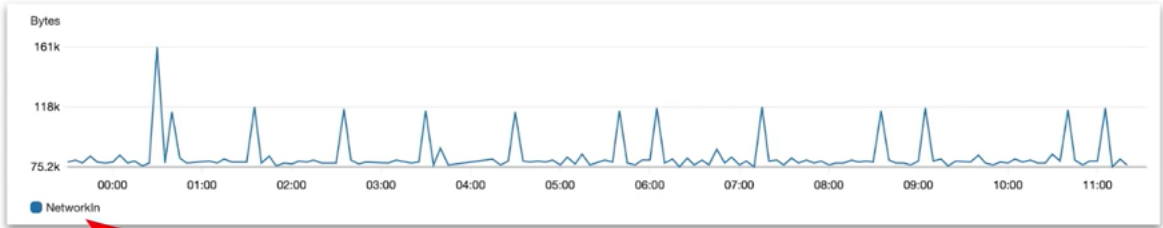
CloudWatch Metrics

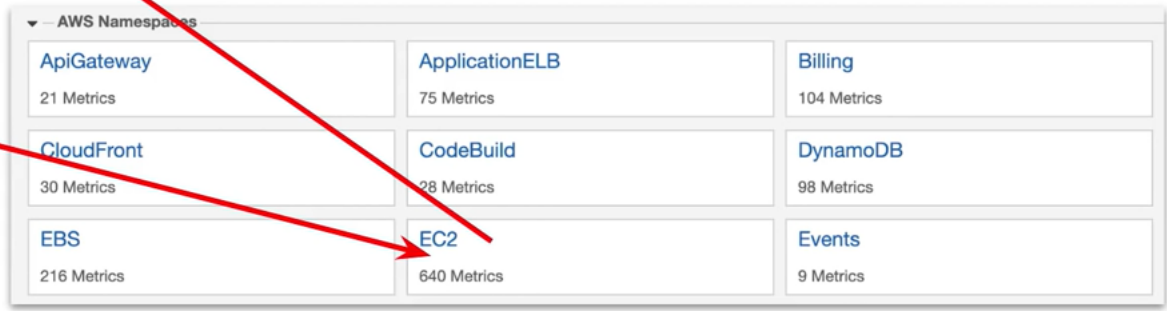
Represents a time-ordered set of data points
A variable to monitor

CloudWatch comes with many **predefined** metrics eg.

EC2 Per-Instance Metrics

- CPUUtilization
- DiskReadOps
- DiskWriteOps
- DiskReadBytes
- DiskWriteBytes
- **NetworkIn**
- NetworkOut
- NetworkPacketsIn
- NetworkPacketsOut





amazon web services

AWS Solutions Architect Associate

CloudWatch

CloudWatch Events

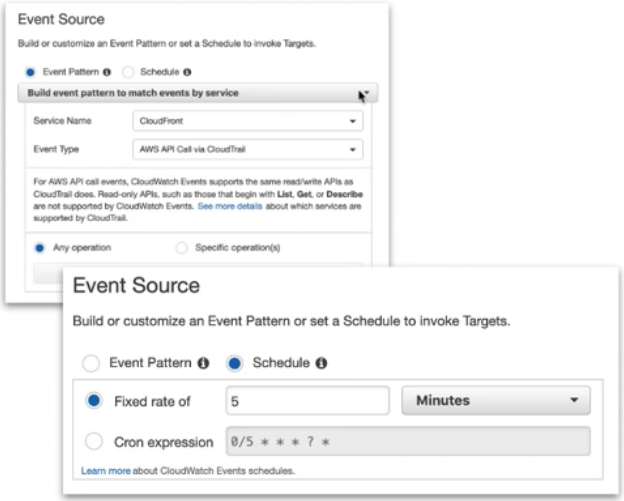
amazon web services

CloudWatch Events

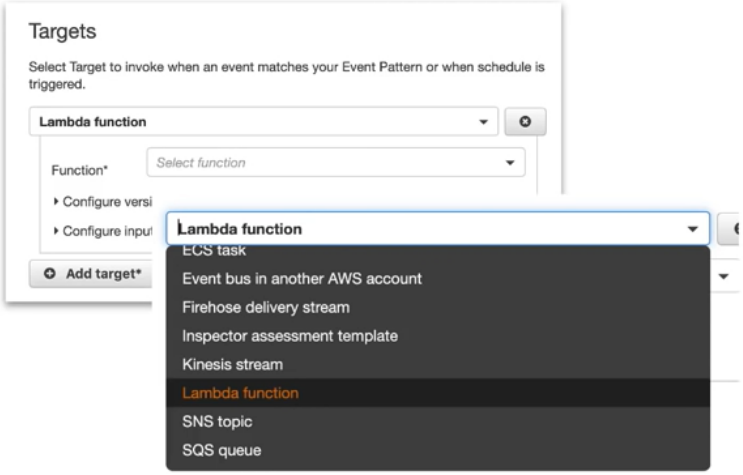
Trigger an event based on a condition or on schedule

Event Source
How to trigger the event

Targets
What to trigger



Schedule is like a serverless **Cron tab**



This is a very big list

amazon web services

AWS Solutions Architect Associate

CloudWatch

CloudWatch Custom Metrics

amazon web services

Custom Metrics and High Resolution Metrics

Using the **AWS CLI or SDK** you can create and publish your own **custom metrics**.

```
aws cloudwatch put-metric-data \
    --metric-name Enterprise-D  \
    --namespace Starfleet \
    --unit Bytes \
    --value 231434333 \
    --dimensions HullIntegrity=100,Shield=70,Thrusters=maximum
```

High Resolution Metrics

When you publish a custom metric, you can define it as either standard resolution or **high resolution**

High resolution lets you track **under 1 minute down** to **1 second**.

With High Resolution you can track at:

- **1 second**
- 5 seconds
- 10 seconds
- 30 seconds
- multiple of 60 seconds.

AWS Solutions Architect Associate

CloudWatch

CloudWatch Alarms

amazon web services

CloudWatch Alarms

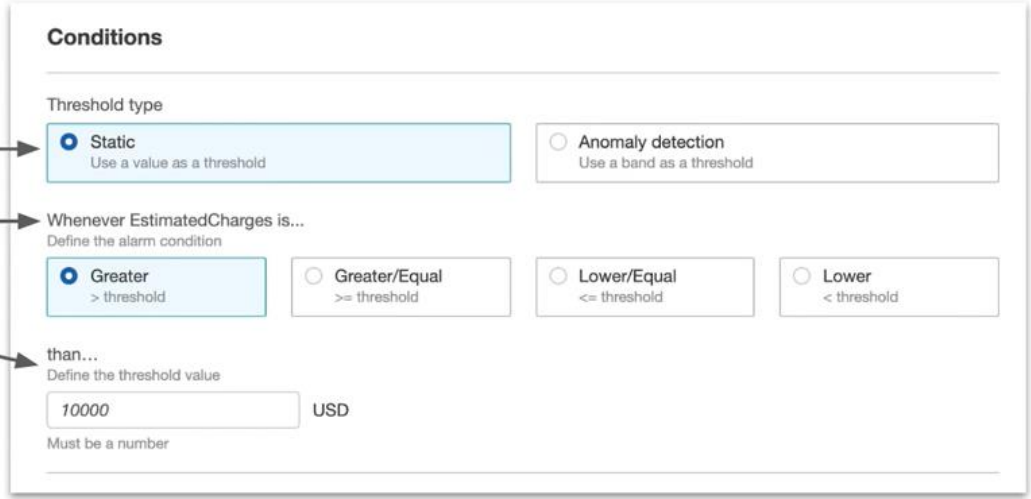
Triggers a notifications based on **metrics** which **breach** a defined threshold

☐	1000 USD Billing	OK	EstimatedCharges >= 1000 for 1 datapoints within 6 hours

The **Type**

The **Condition**

The **Threshold**



amazon web services

AWS Solutions Architect Associate

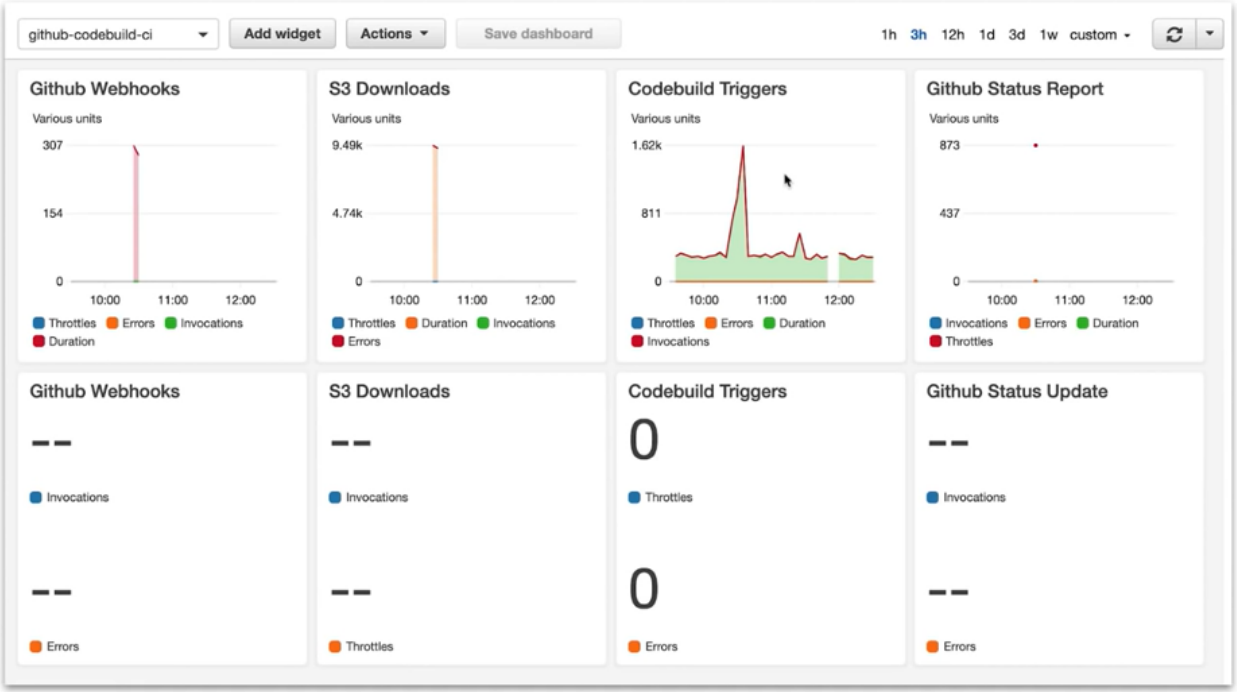
CloudWatch

CloudWatch Dashboards

amazon web services

CloudWatch Dashboards

Create **custom dashboards** from CloudWatch Metrics



amazon web services

AWS Solutions Architect Associate

CloudWatch

CloudWatch Availability

amazon web services

CloudWatch – Availability of Data

How often CloudWatch will collect and make available data.

	EC2	Other services
Basic Monitoring	5 minute interval	**1 minute** / 3 minute / 5 minute
Detailed Monitoring	1 minute interval	

Most services are 1 minute by default

AWS Solutions Architect Associate

CloudWatch

Agent & Host Level Metrics

amazon web services

CloudWatch Agent and Host Level Metrics

Some metrics you might think are tracked by default for EC2 instances are not, and require install the **CloudWatch Agent**.

The CloudWatch Agent is a script which can be installed via

Systems Manager Run Command onto the target EC2 instance.

CloudWatch will track at the Host Level by default:

- CPU Usage
- Network Usage
- Disk Usage
- Status Checks
  - Underlying Hypervisor status
  - Underlying EC2 instance status

The following require the Agent to get detailed metrics for:

- **Memory** utilization
- Disk swap utilization
- **Disk space** utilization
- Page file utilization
- Log collection

AWS Solutions Architect Associate

CloudWatch

CloudWatch Cheat Sheet

amazon web services

CloudWatch CheatSheet

- CloudWatch is a collection of monitoring services: **Dashboards**, **Events**, **Alarms**, **Logs** and **Metrics**
- CloudWatch **Logs:** log data from AWS services. eg. CPU Utilization
- CloudWatch **Metrics:** Represents a time-ordered set of data points, A variable to monitor eg. CPU Utilization over time
- CloudWatch **Events:** trigger an event based on a condition eg. ever hour take snapshot of server
- CloudWatch **Alarms**: triggers notifications based on metrics when a defined threshold is breached
- CloudWatch **Dashboards**: create visualizations based on metrics
- EC2 monitors at 5 min intervals and at Detailed Monitoring 1 minute intervals
- Most other service monitor at 1 minute intervals, with intervals of 1 , 3 , 5 minutes.
- Logs must belong to a **Log Group**
- CloudWatch Agent needs to be installed on EC2 host to track **Memory Usage** and **Disk Size**
- You can can stream custom log files eg. production.log
- Custom Metrics allow you to track High Resolution Metrics a sub minute intervals all the way down to 1 second.

AWS Products for Deployment and Management

- Amazon CloudWatch 
- Amazon Elastic Beanstalk 
- AWS CloudFormation 

AWS Solutions Architect Associate

Elastic Beanstalk

Elastic Beanstalk Introduction

amazon web services

Elastic Beanstalk

Quickly **deploy and manage** web-apps on AWS
without worrying about infrastructure.

amazon web services

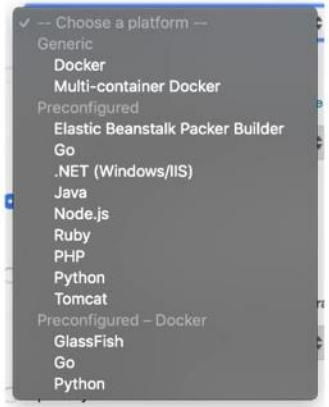
Elastic Beanstalk

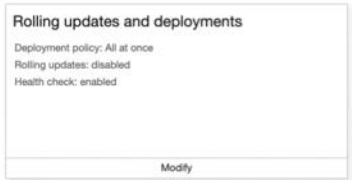
The Heroku of AWS. Choose a platform, upload your code and it runs with little worry for developers about infrastructure knowledge. **Not Recommended for "Production" applications**

Elastic Beanstalk is powered by a CloudFormation template setups for you:

- Elastic Load Balancer
- Autoscaling Groups
- RDS Database
- EC2 Instance preconfigured (or custom) platforms
- Monitoring (CloudWatch, SNS)
- In-Place and Blue/Green deployment methodologies
- Security (Rotates passwords)
- Can run **Dockerized** environments

AWS is talking about enterprise, large companies.





amazon web services

AWS Solutions Architect Associate

Elastic Beanstalk

Elastic Beanstalk Cheat Sheet

amazon web services

Elastic Beanstalk *CheatSheet*

- **Elastic Beanstalk** handles the deployment, from capacity provisioning, load balancing, auto-scaling to application health monitoring
- When you want to run a web-application but you don't want to have think about the underlying infrastructure.
- It costs nothing to use Elastic Beanstalk (only the resources it provisions eg. RDS, ELB, EC2)
- Recommended for test or development apps. Not recommended for production use
- You can choose from the following preconfigured platforms: Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker
- You can run dockerized environments on Elastic Beanstalk.

AWS Products for Deployment and Management

- Amazon CloudWatch 
- Amazon Elastic Beanstalk 
- AWS CloudFormation 

AWS Solutions Architect Associate

CloudFormation

Introduction to CloudFormation

amazon web services

AWS CloudFormation

A Templating Language that **defines AWS resources** to be provisioned.
Automating the creation of resources via code.

***Infrastructure as Code**

CloudFormation – Introduction

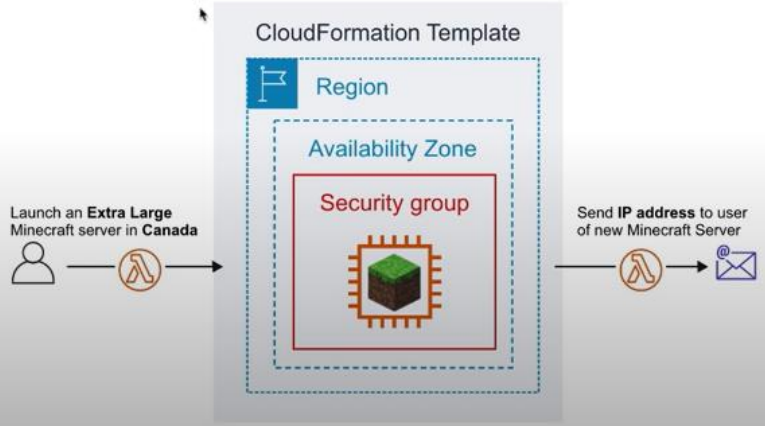
What is Infrastructure As Code? (IaC)

the process of managing and provisioning computer data centers (eg, AWS) through machine-readable definition files (eg, YAML, JSON files) rather than physical hardware configuration or interactive configuration tools. (stop doing manual configuration!)

Use Case

People pay a monthly subscription and we run a Minecraft server. They choose **where** they want and **what size** of server they want to run.

We can use their **inputs** and use an AWS Lambda to create a new CloudFormation stack. We have a lambda send them the email of their new Minecraft Server IP address and details.



amazon web services

AWS Solutions Architect Associate

CloudFormation

Template Formats

amazon web services

CloudFormation – Template Formats

CloudFormation can be written in ✌️ two different formats:

👉 JSON

YAML 👉

AWS Solutions Architect Associate

CloudFormation

Template Anatomy

amazon web services

CloudFormation – Template Anatomy

```yaml
AWSTemplateFormatVersion: 2010-09-09
Description: >-
  Launch an EC2 Instance running apache and expose
  default page to the internet. Harcoded to work
  only with  US-EAST-1 (N. Virginia)
Parameters:
  InstanceType:
    Description: WebServer EC2 instance type
    Type: String
    Default: t2.micro
    AllowedValues:
      - t2.nano
      - t2.micro
Resources:
  WebServer:
    Type: 'AWS::EC2::Instance'
    Properties:
      Tags:
        -
          Key: Name
          Value: Apache Default WebServer
      InstanceType: !Ref InstanceType
      # You shouldn't hardcode, just shown here for example
      ImageId: 'ami-0b898040803850657'
      SecurityGroupIds:
        - !GetAtt SecurityGroup.GroupId
      UserData:
        'Fn::Base64':
          !Sub |
          #!/usr/bin/env bash
          su ec2-user
          sudo yum install httpd -y
          sudo service httpd start
  SecurityGroup:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      GroupDescription: Enable internet users to access.
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: 0.0.0.0/0
Outputs:
  PublicIp:
```

Template Sections

MetaData Additional information about the template

Description A description of what this template is suppose to do

Parameters Values to pass to your template at runtime

Mappings A lookup table. Maps keys to values so you change your va to something else

Conditions Whether resources are created or properties are assigned

Transform Applies marcos (like applying a mod which change the ana to be custom)

Resources* A resource you want to create eg. IAM Role, EC2 Instance, Lambda, RDS

Outputs Values that returned eg. an ip-address of new server creat

CloudFormation Templates **requires** you to **at least list one resource**.

amazon web services

AWS Solutions Architect Associate

CloudFormation

AWS Quick Starts

amazon web services

AWS Quick Starts

AWS Quick Starts are a collection of **pre-built CloudFormation templates**.

- Analytics
- Blockchain
- Business productivity
- Communications
- Contact center
- Containers & microservices
- Data lakes
- Databases
- DevOps
- Healthcare
- Infrastructure
- IoT
- Life sciences
- Machine learning & AI
- Media services
- Migration
- Networking & remote access
- SaaS
- Security, identity, compliance
- Serverless
- Storage
- Websites & web apps
- IBM
- Microsoft
- SAP

IOT | SERVERLESS

Quick Start

ONICA

AWS IoT Camera Connector

Built by Onica and AWS

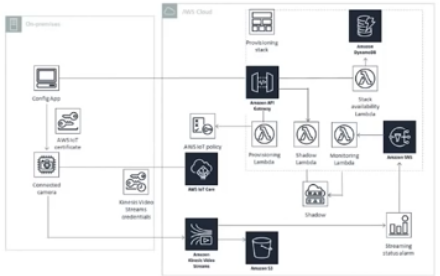
Builds a serverless architecture to connect and manage cameras through AWS IoT Core, and to stream camera

Time to deploy
5 min

What you'll build | **How to deploy** | **Cost and licenses**

Use this Quick Start to set up the following **serverless** architecture on AWS:

- An AWS IoT policy to associate with connected cameras.
- An AWS Identity and Access Management (IAM) role for connected cameras to stream to Kinesis Video Streams.
- An Amazon DynamoDB table to store provisioning keys. Once provisioning is complete, you should remove the keys.
- AWS Lambda functions to create a provisioning key and a role alias, verify the stack, and provision cameras.
- Amazon API Gateway to expose provisioning endpoints through HTTPS.
- Amazon CloudWatch alarms to expose camera streaming status through an Amazon Simple Notification Service (Amazon SNS) topic, and update the associated camera's IoT thing shadow.
- A separate Config App installable for provisioning cameras on the local network to stream to your AWS account.



Switch to full-screen view

amazon web services

AWS Solutions Architect Associate

CloudFormation

CloudFormation Cheat Sheet

amazon web services

CloudFormation *CheatSheet*

- When being asked to **automate** the provisioning of resources *think* CloudFormation
- When Infrastructure as Code (IaC) is mentioned *think* CloudFormation
- CloudFormation can be written in either JSON or YAML
- When CloudFormation encounters an error it will rollback with **ROLLBACK_IN_PROGRESS**
- CloudFormation templates larger than 51,200 bytes (0.05 MB) are too large to upload directly, and must be imported into CloudFormation via an S3 bucket.
- **NestedStacks** helps you break up your CloudFormation template into smaller reusable templates that can be composed into larger templates
- **At least one resource** under resources: must be defined for a CloudFormation template **to be valid**
- **MetaData** extra information about your template
- **Description** a description of what the template is suppose to do
- **Parameters** is how you get user inputs into templates
- **Transforms** Applies marcos (like applying a mod which change the anatomy to be custom)
- **Outputs** are values you can use to import into other stacks
- **Mappings** maps keys to values, just like a lookup table
- **Resources** defines the resources you want to provision, **at least one resource is required**
- **Conditions** are whether resources are created or properties are assigned

amazon web services

AWS Solutions Architect Associate

amazon web services

CloudTrail

Logs API calls between AWS services.
When you need to know who to blame.

amazon web services

Introduction to CloudTrail

AWS CloudTrail is a service that enables **governance**, **compliance**, **operational auditing**, and **risk auditing** of your AWS account.

AWS CloudTrail is used to **monitor API calls** and **Actions** made on an AWS account.

Easily identify which users and accounts made the call to AWS eg.

- **Where** Source IP Address
- **When** EventTime
- **Who** User, UserAgent
- **What** Region, Resource, Action


```
{"Records": [{
    "eventVersion": "1.0",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:user/Worf",
        "accountId": "123456789012",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Worf"
    },
    "eventTime": "2014-03-24T21:11:59Z",
    "eventSource": "iam.amazonaws.com",
    "eventName": "CreateUser",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-cli/1.3.2 Python/2.7.5 Windows/10",
    "requestParameters": {"userName": "LaForge"},
    "responseElements": {"user": {
        "createDate": "Mar 24, 2014 9:11:59 PM",
        "userName": "LaForge",
        "arn": "arn:aws:iam::123456789012:user/LaForge",
        "path": "/",
        "userId": "EXAMPLEUSERID"
    }}
}]}
```

AWS Solutions Architect Associate

CloudTrail

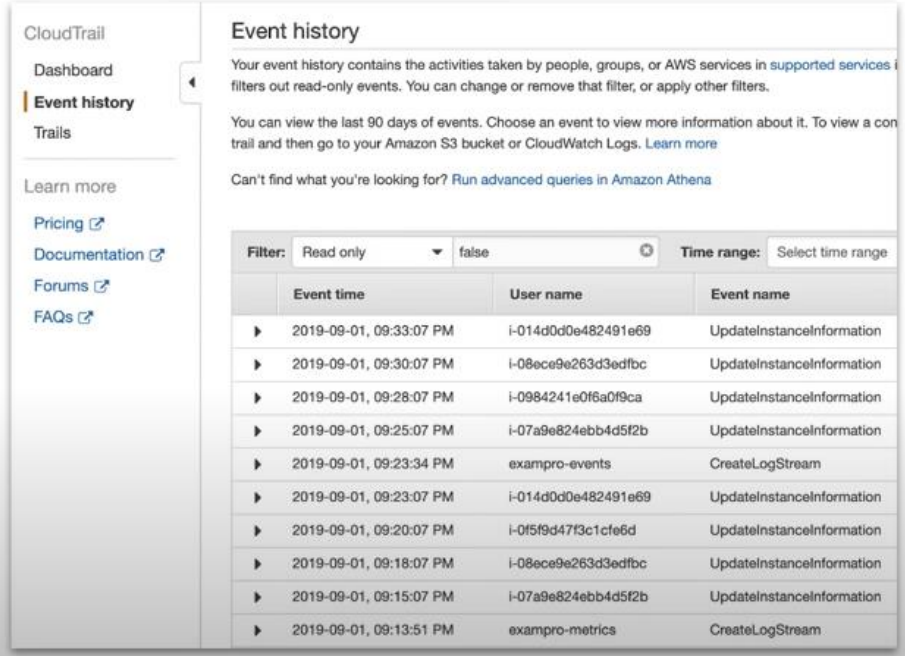
CloudTrail Event History

CloudTrail - Event History

CloudTrail is already logging by default and will collect logs for **last 90 days** via **Event History**

If you need more than 90 days you need to create a **Trail**

Trails are output to S3 and do not have GUI like Event History. To analyze a Trail you'd have to use **Amazon Athena.**



AWS Solutions Architect Associate

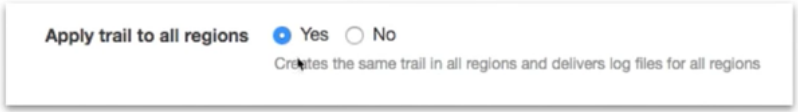
CloudTrail

CloudTrail
Trail Options

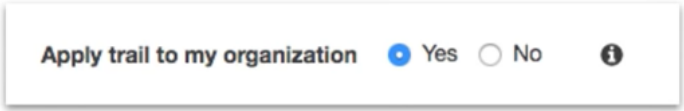
amazon web services

CloudTrail – Trail Options

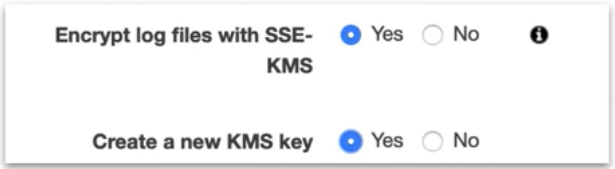
A Trail can be set to **log to all regions**



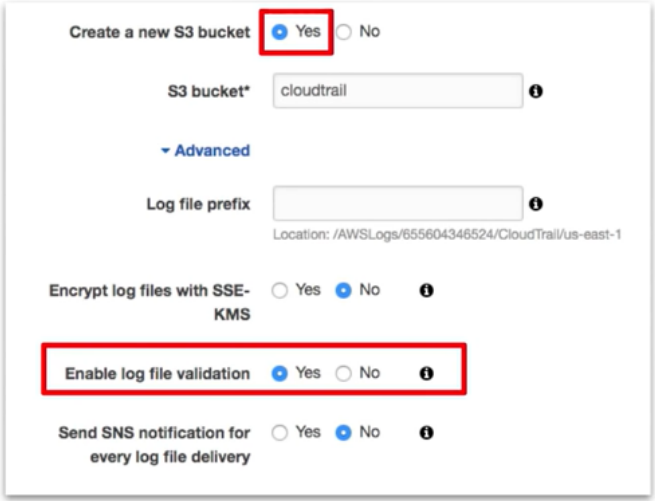
A Trail can be set to **across all accounts in an Organization**



You can **Encrypt your Logs** using Server Side Encryption via Key Management Service (SSE-KMS)



We can ensure the Integrity of our logs to see if they have been tampered we need to turn on **Log File Validation**



AWS Solutions Architect Associate

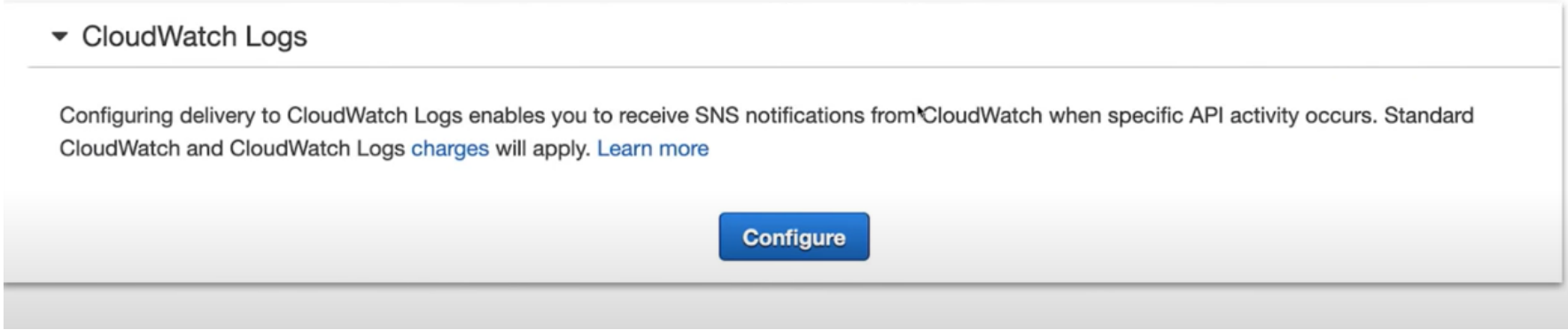
CloudTrail

CloudTrail to CloudWatch

amazon web services

CloudTrail to CloudWatch

CloudTrail can be set to deliver events to a CloudWatch log.



AWS Solutions Architect Associate

CloudTrail

Management vs Data Events

amazon web services

CloudTrail – Management Events vs Data Events

Management Events

Tracks management operations. Turned on by default. Can't be turned off.

- **Configuring security**

  eg. IAM AttachRolePolicy API operations
- **Registering devices**

  eg. Amazon EC2 CreateDefaultVpc API operations)
- **Configuring rules for routing data**

  eg. Amazon EC2 CreateSubnet API operations
- **Setting up logging**

  eg. AWS CloudTrail CreateTrail API operations

Data Events

Tracks specific operations for specific AWS Services. Data events are high volume logging and will result in additional charges. **Turned off by default**

The two services that can be tracked is S3 and Lambda. So it would track action such as: GetObject, DeleteObject, PutObject

AWS Solutions Architect Associate

CloudTrail

CloudTrail Cheat Sheet

amazon web services

CloudTrail *CheatSheet*

- CloudTrail logs calls between AWS services
- **governance**, **compliance**, **operational auditing**, and **risk auditing** are keywords relating to CloudTrail
- When you need to know **who to blame** *think* CloudTrail
- CloudTrail by default logs event data for the past 90s days via **Event History**
- To track beyond 90 days you need to create **Trail**
- To ensure logs have not been tampered with you need to turn on **Log File Validation** option
- CloudTrail logs can be encrypted using **KMS (Key Management Service)**
- CloudTrail can be set to log across all AWS accounts in an Organization and all regions in an account.
- CloudTrail logs can be streamed to CloudWatch logs
- Trails are outputted to an S3 bucket that you specify
- CloudTrail logs two kinds of events: **Management Events** and **Data Events**
- **Management events** log management operations eg. AttachRolePolicy
- **Data Events** log data operations for resources (S3, Lambda) eg. GetObject, DeleteObject, and PutObject
- Data Events are **disabled** by default when creating a Trail.
- Trail logs in S3 can be analyzed using Athena

Module review

- List the three main AWS products for network and content delivery
- List the three main AWS products for deployment and management