

Module 6: Overview of Services for Web Applications

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

Domain Name System - DNS



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



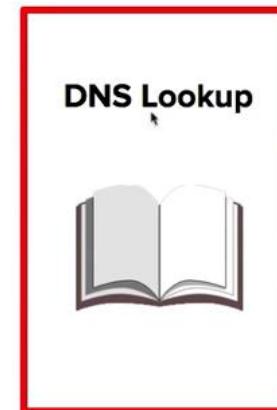
Introduction to DNS

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)

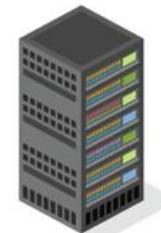
This is what allows your computer to **find specific servers** on the internet automatically **depending what domain name** you browse to.



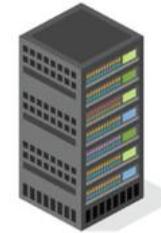
www.exampro.co



52.216.8.34



55.211.9.39





AWS Solutions Architect Associate

DNS

Internet Protocol (IP)



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

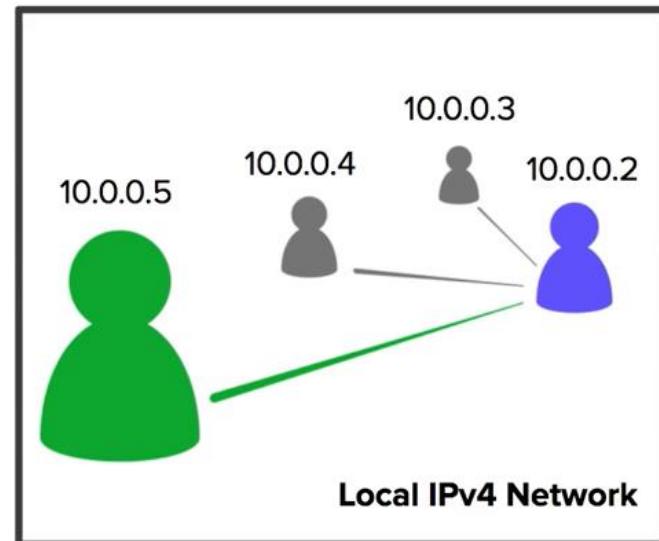
Example: **52.216.8.34**

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



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**.

The screenshot shows the WhoIS search results for the domain `exampro.co`. The page has a header with the WhoIS logo and navigation links for DOMAINS, WEBSITE, CLOUD, HOSTING, SERVERS, EMAIL, SECURITY, and WHOIS. Below the header, it says "exampro.co Updated 1 second ago". The main content area is divided into sections: "Domain Information" and "Registrant Contact".
Domain Information:
Domain: exampro.co
Registrar: CCI REG S.A.
Registered On: 2018-05-21
Expires On: 2020-05-21
Updated On: 2018-09-20
Status: clientTransferProhibited
Name Servers: ns-568.awsdns-07.net, ns-1965.awsdns-53.co.uk, ns-415.awsdns-51.com, ns-1027.awsdns-00.org
Registrant Contact:
State: ON
Country: CA

Some Popular Domain Registrars You May Know...





AWS Solutions Architect Associate

DNS

Top-Level Domains



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 known 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>



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

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 Solutions Architect Associate

DNS

Start of Authority (SOA)



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

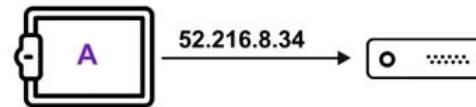


Address Records

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



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": "www.testing-domain.com"}  
      ]  
    }  
  ]  
}
```



AWS Solutions Architect Associate

DNS

NS Records



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."},  
      ]  
    }  
  ]  
}
```



AWS Solutions Architect Associate

DNS

Time To Live (TTL)



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 user's 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

DNS



DNS Cheat Sheet



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

Route53



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



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

Choose a domain name





AWS Solutions Architect Associate

Route53



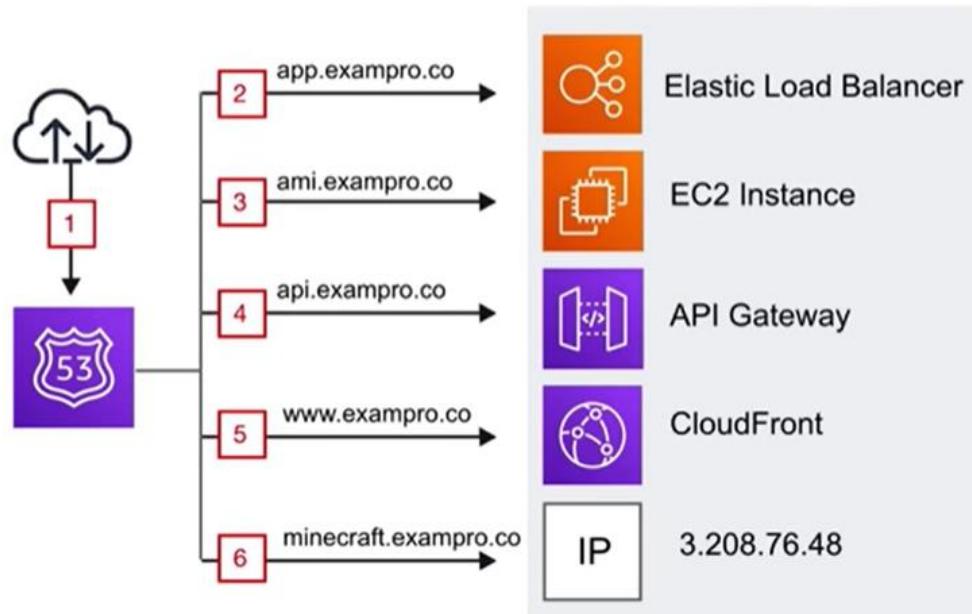
Route53 Use Case



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



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.

Create Record Set

Name: www.exampro.co.

Type: A – IPv4 address
 CNAME – Canonical name
 MX – Mail exchange
 AAAA – IPv6 address
 TXT – Text
 PTR – Pointer
 SRV – Service
 SPF – Sender Policy Framework
 NAPTR – Network Address and Port Translation
 CAA – Certificate Authority Authorization
 NS – Name Server
 SOA – Start of Authority
198.51.100.234

Alias: Yes No

TTL (Seconds): 300 1m 5m 1h 1d

Value: 192.0.2.235

IPv4 address. Enter multiple addresses on separate lines.
Example:
192.0.2.235
198.51.100.234



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

Create Record Set

Name: www .exampro.co.

Type: A – IPv4 address

Alias: Yes No

Alias Target:

You can also type — S3 website endpoints —

- CloudFront distribution domain name: d11111abcdef8.cloudfront.net
- Elastic Beanstalk environment CNAME: example.elasticbeanstalk.com
- ELB load balancer DNS name: example-1.us-east-2.elb.amazonaws.com
- S3 website endpoint: s3-website.us-east-2.amazonaws.com
- Resource record set in this hosted zone: www.example.com
- VPC endpoint: example.us-east-2.vpc.amazonaws.com
- API Gateway custom regional API: d-abcde12345.execute-api.us-west-2.amazonaws.com





AWS Solutions Architect Associate

Route53



Routing Policies



Route53 - Routing Policies

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

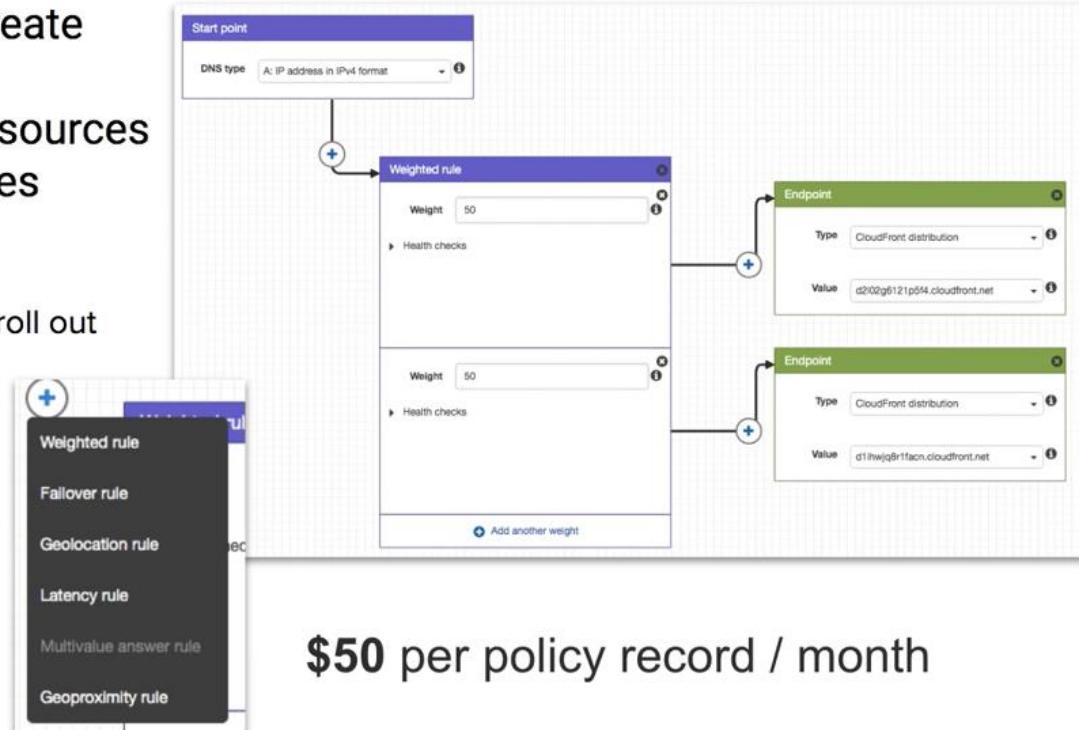
- **Simple Routing** default routing policy, multiple addresses result in random selection
- **Weighted Routing** route traffic based on weighted values to split traffic
- **Latency-Based Routing** route traffic to region resource with lowest latency
- **Failover Routing** route traffic if primary endpoint 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.





AWS Solutions Architect Associate

Route53



Simple Routing Policies



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.

www.exampro.co

Create Record Set

Name: random.exampro.co.

Type: A – IPv4 address

Alias: Yes No

TTL (Seconds): 300 1m 5m 1h 1d

Value:

34.229.79.211
18.212.245.88
3.208.76.58

IPv4 address. Enter multiple addresses on separate lines.
Example:
192.0.2.235
198.51.100.234

Routing Policy: Simple

Route 53 responds to queries based only on the values in this record. [Learn More](#)

Simple Policy



34.229.79.211
18.212.245.88
→ 3.208.76.48



AWS Solutions Architect Associate

Route53



Weighted Routing Policies

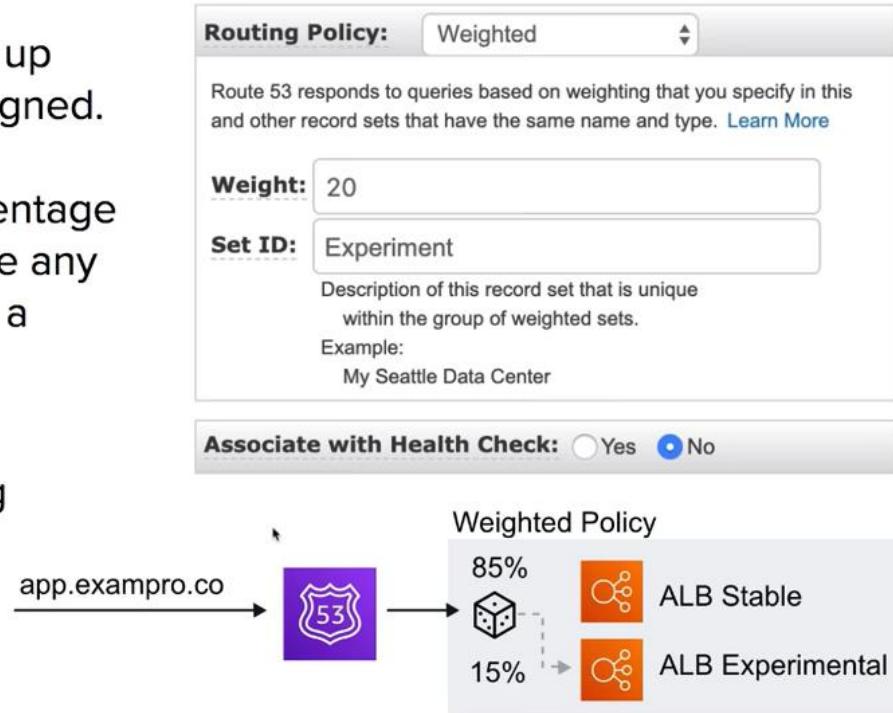


Route53 - Weighted Routing Policies

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



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. A request comes in from Toronto, Canada. It will be routed to Montreal since it will have lower latency

app.exampro.co
Toronto, Canada

Routing Policy: Latency

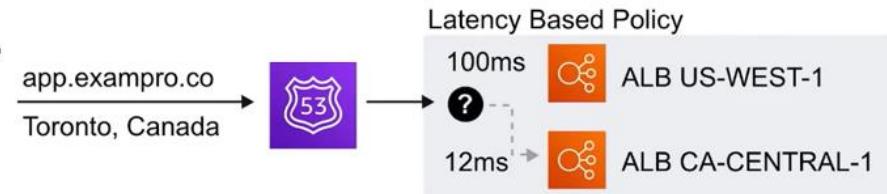
Route 53 responds to queries based on regions that you specify in this and other record sets that have the same name and type. [Learn More](#)

Region: us-east-1

Set ID: Fast Zone

Description of this record set that is unique within the group of latency sets.
Example:
My Seattle Data Center

Associate with Health Check: Yes No





AWS Solutions Architect Associate

Route53



Failover Routing Policies



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.

Routing Policy: Failover

Route 53 responds to queries using primary record sets if any are healthy, or using secondary record sets otherwise. [Learn More](#)

Failover Record Type: Primary Secondary

Set ID: prod-Primary

Associate with Health Check: Yes No





AWS Solutions Architect Associate

Route53



Geolocation Routing Policies



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.

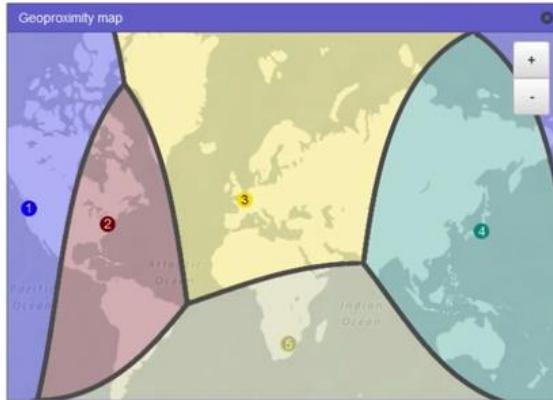
You must use Route53 Traffic Flow in order to use geoproximity routing policies

The screenshot shows the 'Geoproximity rule' configuration page. It includes fields for 'Region' (set to '1'), 'Endpoint Location' (set to 'US East (N. Virginia)'), 'Coordinates' (set to 'Using US East (N. Virginia) Coordinates'), and a 'Bias' slider set to '-2'. There is also a 'Health checks' section.

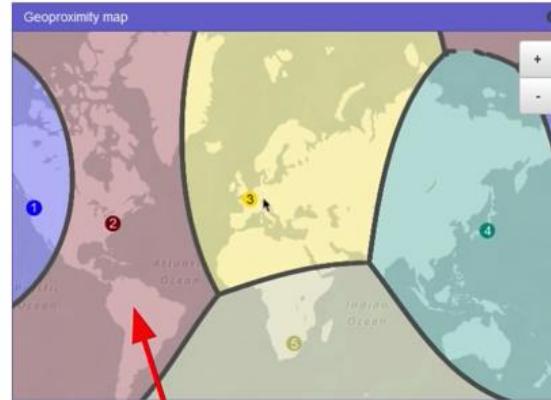


Route53 – Geoproximity Routing Policies

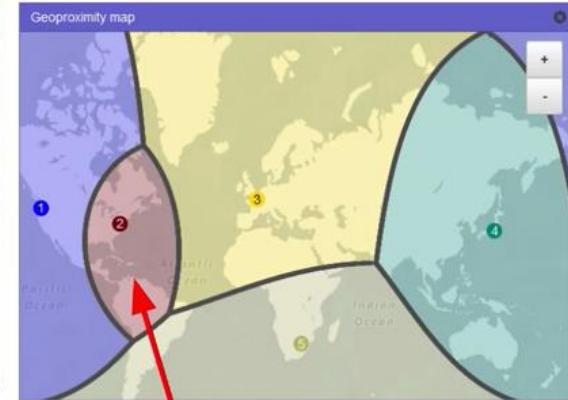
How Bias Works for Geoproximity



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

Geoproximity rule

Hide geoproximity map

Region: ①

Endpoint Location

- China (Beijing)
- EU (Frankfurt)
- EU (London)

Coordinates

- EU (Paris)
- EU (Stockholm)

Bias

- Asia Pacific (Singapore)
- Asia Pacific (Sydney)
- Asia Pacific (Seoul)
- Asia Pacific (Tokyo)
- Asia Pacific (Hong Kong)
- Asia Pacific (Mumbai)
- China (Beijing) **②**
- China (Ningxia)
- South America (São Paulo)

Health check

Geoproximity map

+

-

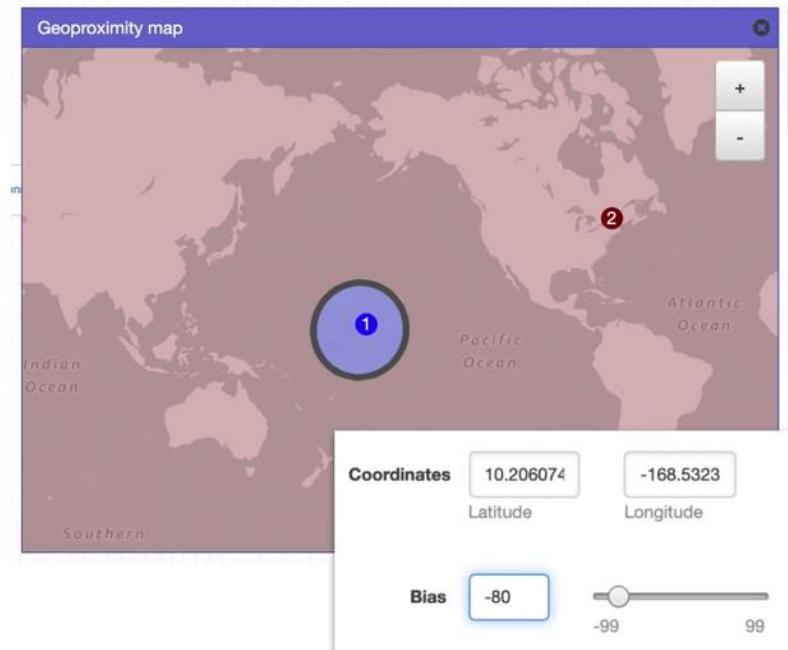


Route53 - Geoproximity Routing Policies

All Current Regions Selected



Can provide **custom coordinates** over region





AWS Solutions Architect Associate

Route53



Multi-Value Answer Policies



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.

Routing Policy: Multivalue Answer

Route 53 responds to DNS queries with up to eight healthy records selected at random. [Learn More](#)

Set ID: multivalue
Description of this record set that is unique within the group of multivalue answer sets.
Example:
Route to Seattle data center

Associate with Health Check: Yes No





AWS Solutions Architect Associate

Route53



Health Checks



Route53 - Health Checks

- Checks health every **30s** by default. Can be reduced to every **10s**
- A health check can **initiate 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

[Create health check](#) [Delete health check](#) [Edit health check](#) [?](#)

Filter by keyword « < 1 to 1 of 1 health check > »

	Name	Status	Description	Alarms
<input type="checkbox"/>	app.exampro.co	<div style="width: 100%; background-color: green; height: 10px;"></div> 15 minutes ago now	Healthy https://app.exampro.co:443/health-check	<input checked="" type="checkbox"/> 1 of 1 in OK



AWS Solutions Architect Associate

Route53



Route53 Resolver



Route53 – Resolver

Formerly 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)

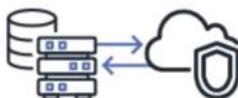
Basic configuration

Direction of DNS queries Info

You can configure endpoints for inbound DNS queries (to your VPC), outbound DNS queries (from your VPC), or both.

Inbound and outbound

Configure endpoints that allow DNS queries both to and from your VPC.



Inbound only

Configure an endpoint that allows DNS queries to your VPC from your network or another VPC.



Outbound only

Configure an endpoint that allows DNS queries from your VPC to your network or another VPC.





AWS Solutions Architect Associate

Route53



Route53 Cheat Sheet



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

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. They 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



ELB - The Rules of Traffic

Listeners

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

Rules (Not available 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 available for Classic Load Balancer)

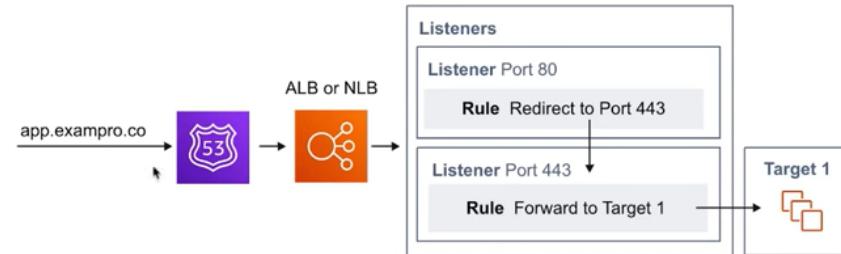
EC2 instances are registered as targets to a Target Group



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.



Description **Listeners** Monitoring Integrated services Tags

A listener checks for connection requests using its configured protocol and port, and the load balancer uses the listener rules to route requests to targets. You can add, remove, or update listeners and listener rules.

Add listener Edit Delete

Listener ID	Security policy	SSL Certificate
HTTP : 80	N/A	N/A
HTTPS : 443	ELBSecurityPolicy	Default: a213d84c-4210-4fc7-3569-9113c39394fb (ACM)

Rules are only for ALB

Rules

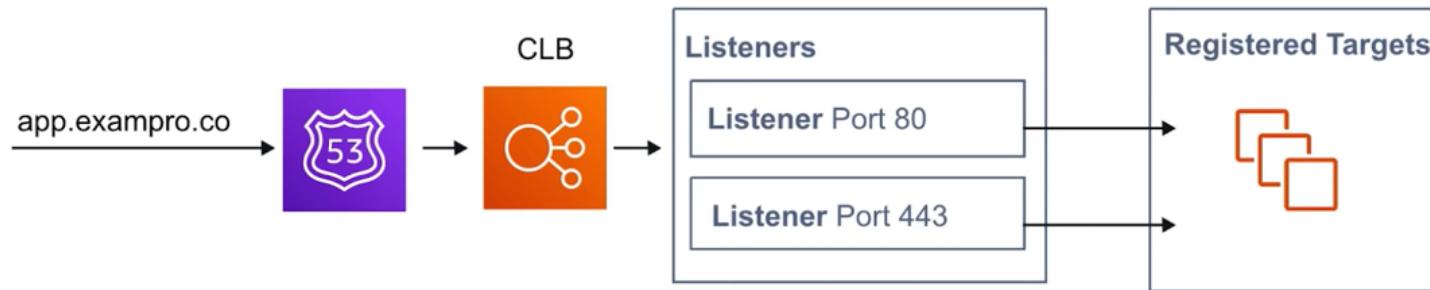
Default: redirecting to HTTPS://#{host}:443/#{path}?#{query}
[View/edit rules](#)

Default: forwarding to production
[View/edit rules](#)



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)



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)



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 perform 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)



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 perform 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



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.**

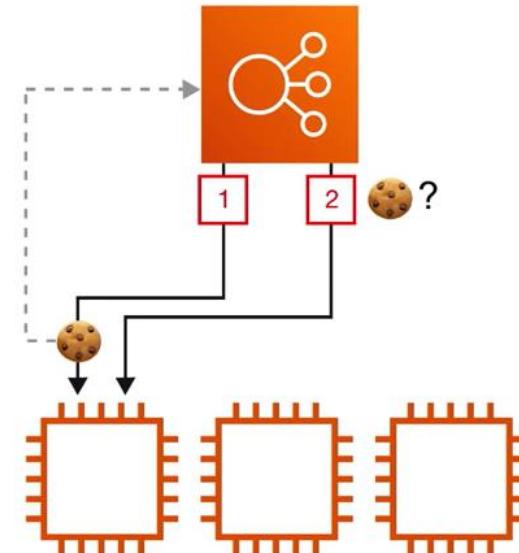
Ensures all **requests** from that session are **sent to the same instance.**

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



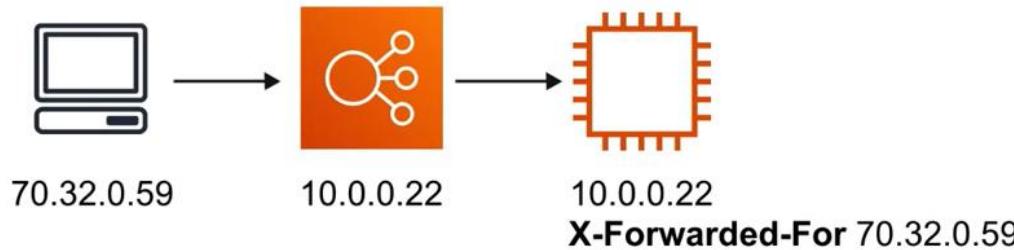
X-Forwarded-For Header



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

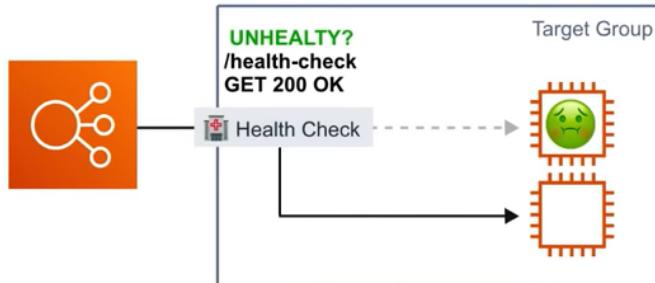


ELB - Health Checks

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 Health checks are found on the **Target Group**

targets	Health checks	Monitoring	Tags
	Protocol i HTTP		
	Path i /health-check		
	Port i traffic port		
	Healthy threshold i 2		
	Unhealthy threshold i 2		
	Timeout i 5		
	Interval i 10		
	Success codes i 200		
		Edit health check	



AWS Solutions Architect Associate

Elastic Load Balancer



Cross-Zone Load Balancing

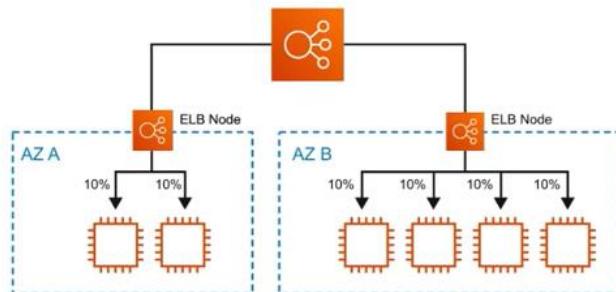


ELB - Cross-Zone Load Balancing

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

Cross-Zone Load Balancing Enabled

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



Attributes

Deletion protection	Disabled
Cross-Zone Load Balancing	Disabled
Access logs	Disabled
<button>Edit attributes</button>	

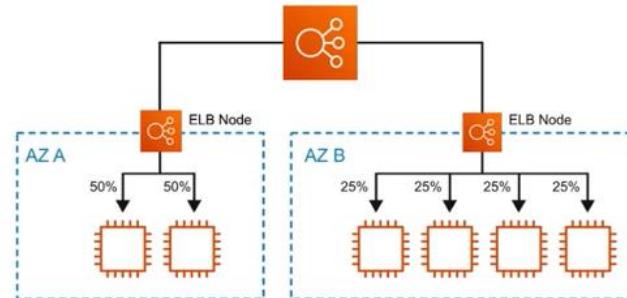
Edit load balancer attributes

Delete Protection	<input type="checkbox"/> Enable
Cross-Zone Load Balancing	<input checked="" type="checkbox"/> Enable

Regional data transfer charges may apply when cross-zone load balancing is enabled. See the documentation for more information.

Cross-Zone Load Balancing Disabled

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





AWS Solutions Architect Associate

Elastic Load Balancer



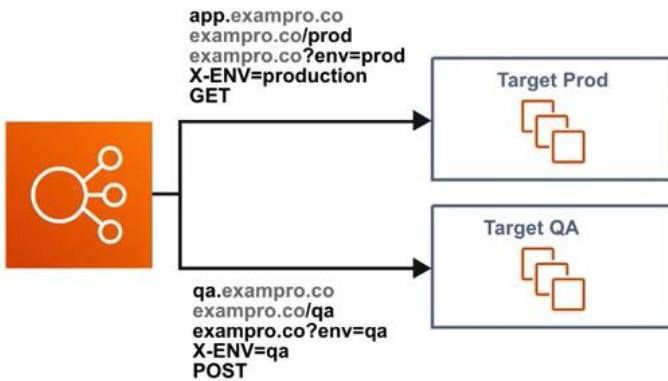
Request Routing



ALB - Request Routing

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

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



The screenshot shows the AWS Lambda Request Rerouting configuration interface with two rules defined:

- Rule 1 (Top):**
 - IF (all match):** Host header..., Path..., Http header..., Http request method..., Query string..., Source IP...
Last: HTTPS 443: default action
This rule cannot be moved or deleted
 - THEN:** Forward to production
- Rule 2 (Bottom):**
 - IF (all match):** Requests otherwise not routed
 - THEN:** Forward to production



AWS Solutions Architect Associate

Elastic Load Balancer



ELB Cheat Sheet



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



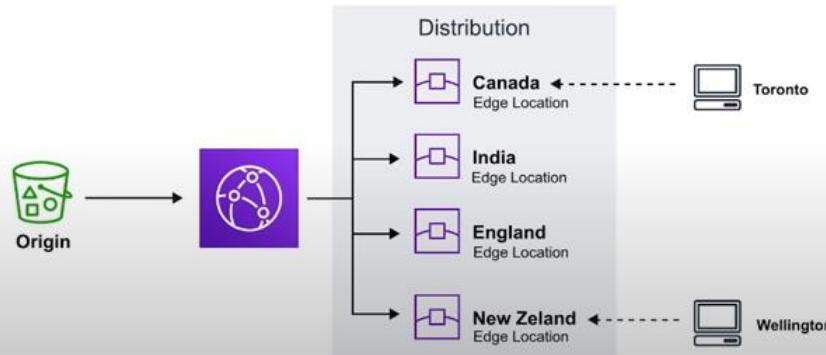
Content Distribution Network (CDN)
Creates cached copies of your website at various
Edge Locations around the world



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

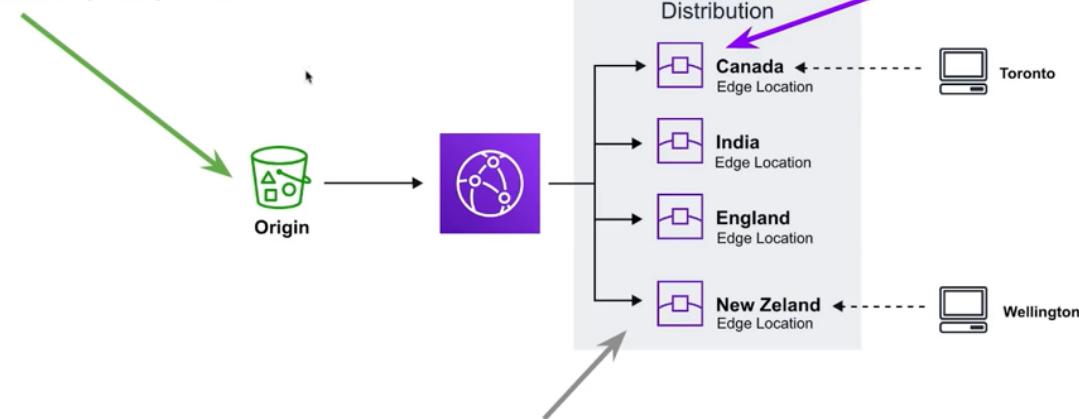


CloudFront Core Components



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



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**

The screenshot shows the 'Distribution Settings' section of the CloudFront console. A dropdown menu for 'Price Class' is open, displaying four options: 'Use All Edge Locations (Best Performance)', 'Use Only U.S., Canada and Europe', 'Use U.S., Canada, Europe, Asia, Middle East and Africa', and 'Use All Edge Locations (Best Performance)'. The second option, 'Use Only U.S., Canada and Europe', is highlighted with a dark background.

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

CloudFront



Lambda@Edge



CloudFront - Lambda@Edge

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

The 4 Available Lambda@Edge Functions

- Viewer request** When CloudFront receives a request from a viewer
- Origin request** Before CloudFront forwards a request to the origin
- Origin response** When CloudFront receives a response from the origin
- Viewer response** Before CloudFront returns the response to the viewer





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

Restrict Viewer Access
(Use Signed URLs or
Signed Cookies)
 Yes
 No

If you restrict viewer access, viewers must use CloudFront signed URLs or signed cookies to access your content. For more information, see Serving Private Content through CloudFront in the Amazon CloudFront Developer Guide.



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



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 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

CloudWatch



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



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

Time (UTC -04:00)	Message
2019-08-31 19:27:29	D, [2019-08-31] [2024-08-06 #13970] DEBUG -- : [0x00000000-0000-0000-0000-000000000000] [1m [34mSELECT * FROM [0x00000000-0000-0000-0000-000000000000] WHERE [0x00000000-0000-0000-0000-000000000000] = '0' [1m [34mUPDATE [0x00000000-0000-0000-0000-000000000000] SET [0x00000000-0000-0000-0000-000000000000] = '1' [1m [34mCOMMIT [0m
19:27:29	D, [2019-08-31]T23:27:29.429028 #13970) DEBUG -- : [0x68d909-7b28-4511-8c27-05050ace10ed] [1m [36mUserMaterial Update (0.8ms) [0m [1m [33mUserIP [0m
19:27:29	D, [2019-08-31]T23:27:29.429028 #13970) DEBUG -- : [0x68d909-7b28-4511-8c27-05050ace10ed] [1m [35m(1.5ms) [0m [1m [35mCOMMIT [0m
19:27:29	D, [2019-08-31]T23:27:29.429028 #13970) DEBUG -- : [0x68d909-7b28-4511-8c27-05050ace10ed] [1m [36mDeck Load (1.0ms) [0m [1m [34mSELECT * FROM [0x00000000-0000-0000-0000-000000000000] WHERE [0x00000000-0000-0000-0000-000000000000] = '0' [1m [34mUPDATE [0x00000000-0000-0000-0000-000000000000] SET [0x00000000-0000-0000-0000-000000000000] = '1' [1m [34mCOMMIT [0m
19:27:29	D, [2019-08-31]T23:27:29.429028 #13970) DEBUG -- : [0x68d909-7b28-4511-8c27-05050ace10ed] [1m [35m(1.1ms) [0m [1m [35mBEGIN [0m
19:27:29	D, [2019-08-31]T23:27:29.429028 #13970) DEBUG -- : [0x68d909-7b28-4511-8c27-05050ace10ed] [1m [36mDeck Load (1.0ms) [0m [1m [34mSELECT * FROM [0x00000000-0000-0000-0000-000000000000] WHERE [0x00000000-0000-0000-0000-000000000000] = '0' [1m [34mUPDATE [0x00000000-0000-0000-0000-000000000000] SET [0x00000000-0000-0000-0000-000000000000] = '1' [1m [34mCOMMIT [0m
19:27:29	D, [2019-08-31]T23:27:29.429028 #13970) DEBUG -- : [0x68d909-7b28-4511-8c27-05050ace10ed] [1m [36mUser Load (0.8ms) [0m [1m [34mSELECT * FROM [0x00000000-0000-0000-0000-000000000000] WHERE [0x00000000-0000-0000-0000-000000000000] = '0' [1m [34mUPDATE [0x00000000-0000-0000-0000-000000000000] SET [0x00000000-0000-0000-0000-000000000000] = '1' [1m [34mCOMMIT [0m
19:27:29	D, [2019-08-31]T23:27:29.429028 #13970) DEBUG -- : [0x68d909-7b28-4511-8c27-05050ace10ed] [1m [36mDeckFlushcard Exists (0.8ms) [0m [1m [34mSELECT * FROM [0x00000000-0000-0000-0000-000000000000] WHERE [0x00000000-0000-0000-0000-000000000000] = '0' [1m [34mUPDATE [0x00000000-0000-0000-0000-000000000000] SET [0x00000000-0000-0000-0000-000000000000] = '1' [1m [34mCOMMIT [0m
19:27:29	D, [2019-08-31]T23:27:29.429028 #13970) DEBUG -- : [0x68d909-7b28-4511-8c27-05050ace10ed] [1m [36mDeckFlushcard Create (0.8ms) [0m [1m [32mIN [0m
19:27:29	D, [2019-08-31]T23:27:29.429028 #13970) DEBUG -- : [0x68d909-7b28-4511-8c27-05050ace10ed] [1m [36mDeckFlushcard Update All (0.8ms) [0m [1m [32mIN [0m
19:27:29	D, [2019-08-31]T23:27:29.429028 #13970) DEBUG -- : [0x68d909-7b28-4511-8c27-05050ace10ed] [1m [36mDeck Update All (0.8ms) [0m [1m [33mUPDATE [0m
19:27:29	D, [2019-08-31]T23:27:29.429028 #13970) DEBUG -- : [0x68d909-7b28-4511-8c27-05050ace10ed] [1m [35m(1.4ms) [0m [1m [35mBEGIN [0m
19:27:29	D, [2019-08-31]T23:27:29.429028 #13970) DEBUG -- : [0x68d909-7b28-4511-8c27-05050ace10ed] [1m [35m(0.8ms) [0m [1m [35mSELECT * FROM [0x00000000-0000-0000-0000-000000000000] WHERE [0x00000000-0000-0000-0000-000000000000] = '0' [1m [34mUPDATE [0x00000000-0000-0000-0000-000000000000] SET [0x00000000-0000-0000-0000-000000000000] = '1' [1m [34mCOMMIT [0m
19:27:29	D, [2019-08-31]T23:27:29.429028 #13970) DEBUG -- : [0x68d909-7b28-4511-8c27-05050ace10ed] [1m [36mFlashcard Load (0.7ms) [0m [1m [34mSELECT * FROM [0x00000000-0000-0000-0000-000000000000] WHERE [0x00000000-0000-0000-0000-000000000000] = '0' [1m [34mUPDATE [0x00000000-0000-0000-0000-000000000000] SET [0x00000000-0000-0000-0000-000000000000] = '1' [1m [34mCOMMIT [0m
19:27:29	D, [2019-08-31]T23:27:29.429028 #13970) DEBUG -- : [0x68d909-7b28-4511-8c27-05050ace10ed] [1m [36mUser Load (0.8ms) [0m [1m [34mSELECT * FROM [0x00000000-0000-0000-0000-000000000000] WHERE [0x00000000-0000-0000-0000-000000000000] = '0' [1m [34mUPDATE [0x00000000-0000-0000-0000-000000000000] SET [0x00000000-0000-0000-0000-000000000000] = '1' [1m [34mCOMMIT [0m
19:27:29	D, [2019-08-31]T23:27:29.429028 #13970) DEBUG -- : [0x68d909-7b28-4511-8c27-05050ace10ed] [1m [36mDeckFlushcard Exists (0.8ms) [0m [1m [34mSELECT * FROM [0x00000000-0000-0000-0000-000000000000] WHERE [0x00000000-0000-0000-0000-000000000000] = '0' [1m [34mUPDATE [0x00000000-0000-0000-0000-000000000000] SET [0x00000000-0000-0000-0000-000000000000] = '1' [1m [34mCOMMIT [0m
19:27:29	D, [2019-08-31]T23:27:29.429028 #13970) DEBUG -- : [0x68d909-7b28-4511-8c27-05050ace10ed] [1m [36mDeckFlushcard Create (0.8ms) [0m [1m [32mIN [0m
19:27:29	D, [2019-08-31]T23:27:29.429028 #13970) DEBUG -- : [0x68d909-7b28-4511-8c27-05050ace10ed] [1m [36mDeckFlushcard Update All (0.8ms) [0m [1m [33mUPDATE [0m
19:27:29	D, [2019-08-31]T23:27:29.429028 #13970) DEBUG -- : [0x68d909-7b28-4511-8c27-05050ace10ed] [1m [35m(1.2ms) [0m [1m [35mCOMMIT [0m

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



AWS Solutions Architect Associate

CloudWatch



CloudWatch Metrics



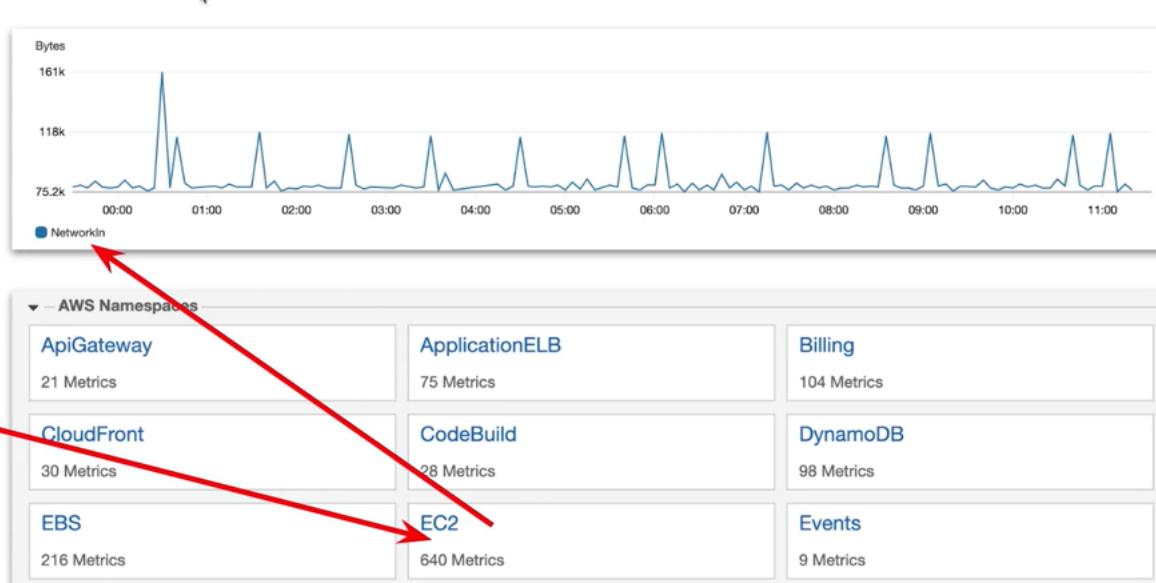
CloudWatch Metrics

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





AWS Solutions Architect Associate

CloudWatch



CloudWatch Events



CloudWatch Events

Trigger an event based on a condition or on schedule

Event Source

How to trigger the event

Event Source
Build or customize an Event Pattern or set a Schedule to invoke Targets.

Event Pattern i Schedule i

Build event pattern to match events by service

Service Name: CloudFront
Event Type: AWS API Call via CloudTrail

For AWS API call events, CloudWatch Events supports the same read/write APIs as CloudTrail does. Read-only APIs, such as those that begin with `List`, `Get`, or `Describe` are not supported by CloudWatch Events. See more details about which services are supported by CloudTrail.

Any operation Specific operation(s)

Event Source
Build or customize an Event Pattern or set a Schedule to invoke Targets.

Event Pattern i Schedule i

Fixed rate of 5 Minutes

Cron expression 0/5 * * * ? *

Learn more about CloudWatch Events schedules.

Schedule is like a serverless **Cron tab**

Targets

What to trigger

Targets

Select Target to invoke when an event matches your Event Pattern or when schedule is triggered.

Lambda function

Function* Select function

Configure versi

Configure input

Add target*

Lambda function

ECS task

Event bus in another AWS account

Firehose delivery stream

Inspector assessment template

Kinesis stream

Lambda function

SNS topic

SQS queue

This is a very big list





AWS Solutions Architect Associate

CloudWatch



CloudWatch Custom Metrics



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



CloudWatch Alarms

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

<input type="checkbox"/> 1000 USD Billing	OK	EstimatedCharges >= 1000 for 1 datapoints within 6 hours
---	----	--

The **Type** → Static
Use a value as a threshold

The **Condition** → Whenever EstimatedCharges is...
Define the alarm condition

The **Threshold** → than...
Define the threshold value

Conditions

Threshold type

Static
Use a value as a threshold

Anomaly detection
Use a band as a threshold

Greater
> threshold

Greater/Equal
>= threshold

Lower/Equal
<= threshold

Lower
< threshold

than...
Define the threshold value

10000 USD
Must be a number



AWS Solutions Architect Associate

CloudWatch

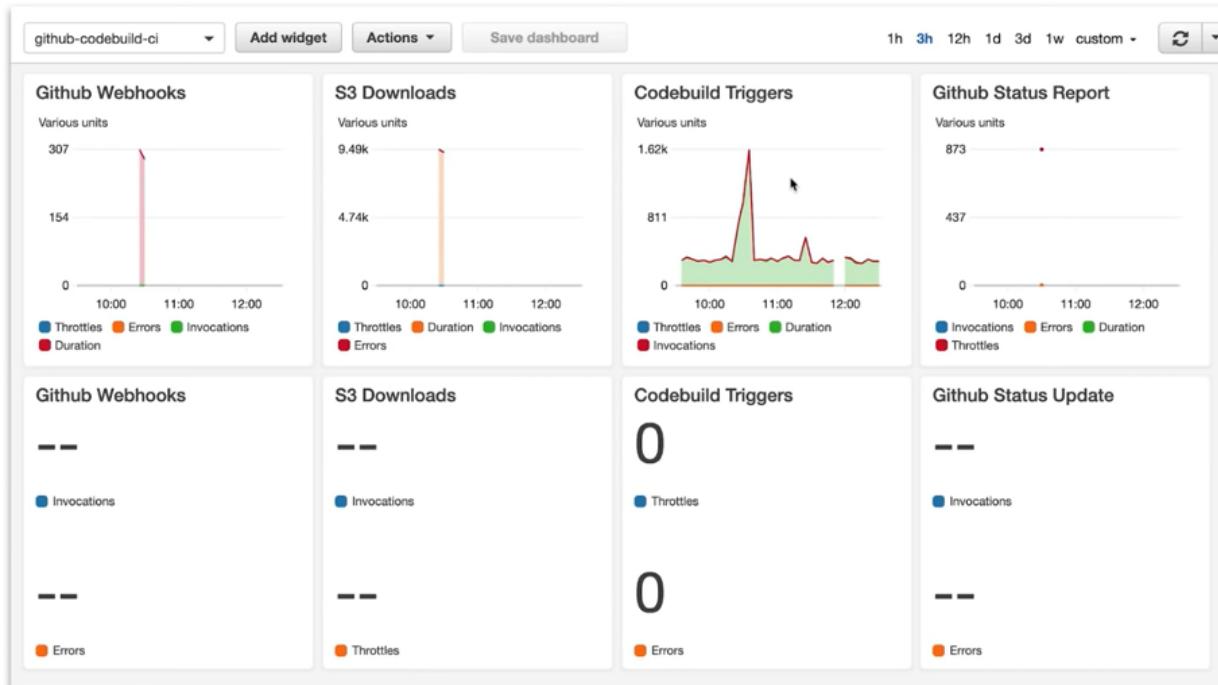


CloudWatch Dashboards



CloudWatch Dashboards

Create **custom dashboards** from CloudWatch Metrics





AWS Solutions Architect Associate

CloudWatch



CloudWatch Availability



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



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



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 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

Elastic Beanstalk



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



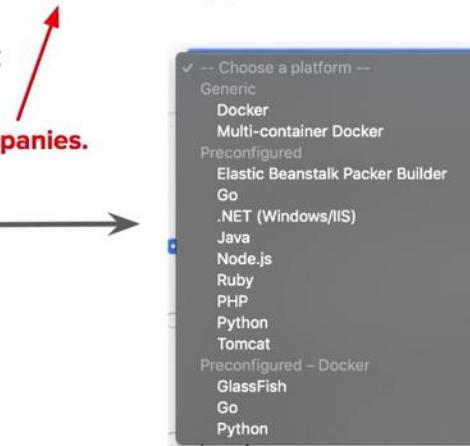
Elastic Beanstalk

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.



Software

AWS X-Ray: disabled
Rotate logs: disabled (default)
Log streaming: disabled (default)
Environment properties: 4
BUNDLE_WITHOUT: RACK_ENV, RAILS_SKIP_ASSET_COMPILATION,
RAILS_SKIP_MIGRATIONS

Modify

Instances

EC2 instance type: t2.micro
EC2 image ID: ami-04cc4d9729e379f0
Root volume type: container default
Root volume size (GB): container default
Root volume iOPS: container default
Security groups: none

Modify

Capacity

Environment type: load balancing, auto scaling
Availability Zones: Any
Instances: 1-4

Modify

Rolling updates and deployments

Deployment policy: All at once
Rolling updates: disabled
Health check: enabled

Modify

Load balancer

Load balancer type: application
Listeners: 1
Processes: 1
Rules: 1

Modify



AWS Solutions Architect Associate

Elastic Beanstalk



Elastic Beanstalk Cheat Sheet



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

AWS CloudFormation



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

***Infrastructure as Code**



CloudFormation – Introduction

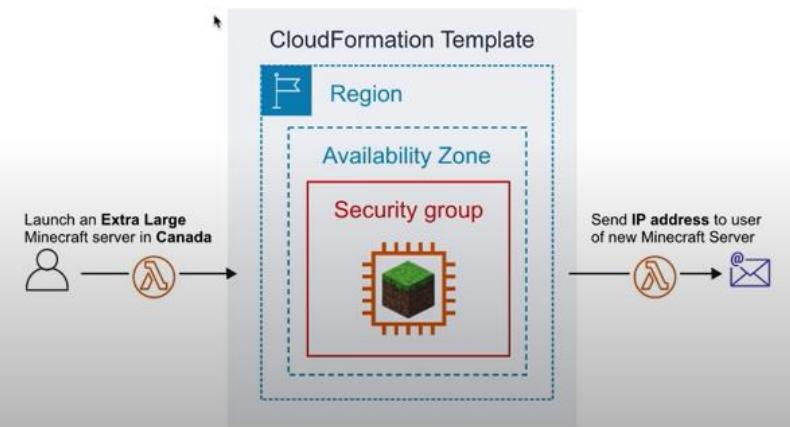
What is Infrastructure As Code? (IaC)

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.





AWS Solutions Architect Associate

CloudFormation



Template Formats



CloudFormation - Template Formats

```
1 {  
2   "AWSTemplateFormatVersion": "2010-09-09",  
3   "Description": "Launch an EC2 Instance running apache and expose default page  
4   "Parameters": {}  
5     "InstanceType": {}  
6       "Description": "WebServer EC2 instance type",  
7       "Type": "String",  
8       "Default": "t2.micro",  
9       "AllowedValues":  
10      "t2.nano",  
11      "t2.micro",  
12    }  
13  }  
14 }  
15 }  
16 "Resources": {  
17   "WebServer": {  
18     "Type": "AWS::EC2::Instance",  
19     "Properties": {}  
20       "Tags": [{}  
21         {"Key": "Name",  
22          "Value": "Apache Default WebServer"}  
23       ]  
24     }  
25     "InstanceType": {}  
26       "Ref": "InstanceType",  
27     }  
28     "ImageId": "ami-0b898040803850657",  
29     "SecurityGroupIds": [{}  
30       {"Fn::GetAtt":  
31         "SecurityGroup",  
32         "GroupId",  
33       }  
34     ]  
35   }  
36   "UserData": {}  
37     "Fn::Base64": {}  
38       "Fn::Sub": "#!/usr/bin/env bash\nsu ec2-user\nsudo yum install httpd -y"  
39     }  
40   }  
41 }  
42 }  
43 }  
44 }  
45 "SecurityGroup": {}  
46   "Type": "AWS::EC2::SecurityGroup",  
47   "Properties": {}  
48     "GroupDescription": "Enable internet users to access.",  
49     "SecurityGroupIngress": [{}  
50       {"IpProtocol": "tcp",  
51         "FromPort": 80,  
52         "ToPort": 80,  
53         "CidrIp": "0.0.0.0/0"}  
54     ]  
55   }  
56 }
```

CloudFormation can be written in 🤝 two different formats:

JSON   YAML

```
1 AWSTemplateFormatVersion: 2010-09-09  
2 Description: -  
3 Launch an EC2 Instance running apache and expose default page to the internet. Hardcoded to work  
4 only with US-EAST-1 (N. Virginia)  
5 Parameters:  
6   InstanceType:  
7     Description: WebServer EC2 instance type  
8     Type: String  
9     Default: t2.micro  
10    AllowedValues:  
11      - t2.nano  
12      - t2.micro  
13 Resources:  
14   WebServer:  
15     Type: 'AWS::EC2::Instance'  
16     Properties:  
17       Tags:  
18         -  
19           Key: Name  
20           Value: Apache Default WebServer  
21     InstanceType: !Ref InstanceType  
22     # You shouldn't hardcode, just shown here for example  
23     ImageId: 'ami-0b898040803850657'  
24     SecurityGroupIds:  
25       - !GetAtt SecurityGroup.GroupId  
26     UserData:  
27       'Fn::Base64':  
28         !Sub  
29           #!/usr/bin/env bash  
30           su ec2-user  
31           sudo yum install httpd -y  
32           sudo service httpd start  
33     SecurityGroup:  
34       Type: 'AWS::EC2::SecurityGroup'  
35       Properties:  
36         GroupDescription: Enable internet users to access.  
37         SecurityGroupIngress:  
38           - IpProtocol: tcp  
39             FromPort: 80  
40             ToPort: 80  
41             CidrIp: 0.0.0.0/0  
42     Outputs:  
43       PublicIp:  
44         Value: !GetAtt WebServer.PublicIp
```



AWS Solutions Architect Associate

CloudFormation



Template Anatomy



CloudFormation - Template Anatomy

```
1 AWSTemplateFormatVersion: 2010-09-09
2 Description: >-
3   Launch an EC2 Instance running apache and expose
4   default page to the internet. Hardcoded to work
5   only with US-EAST-1 (N. Virginia)
6 Parameters:
7   InstanceType:
8     Description: WebServer EC2 instance type
9     Type: String
10    Default: t2.micro
11    AllowedValues:
12      - t2.nano
13      - t2.micro
14 Resources:
15   WebServer:
16     Type: 'AWS::EC2::Instance'
17     Properties:
18       Tags:
19         - Key: Name
20           Value: Apache Default WebServer
21       InstanceType: !Ref InstanceType
22       # You shouldn't hardcode, just shown here for example
23       ImageId: 'ami-0b898040803850657'
24       SecurityGroupIds:
25         - !GetAtt SecurityGroup.GroupId
26     UserData:
27       'Fn::Base64':
28         !Sub |
29           #!/usr/bin/env bash
30           su ec2-user
31           sudo yum install httpd -y
32           sudo service httpd start
33     SecurityGroup:
34       Type: 'AWS::EC2::SecurityGroup'
35       Properties:
36         GroupDescription: Enable internet users to access.
37         SecurityGroupIngress:
38           - IpProtocol: tcp
39             FromPort: 80
40             ToPort: 80
41             CidrIp: 0.0.0.0/0
42     Outputs:
43       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 variable to something else
- Conditions** Whether resources are created or properties are assigned
- Transform** Applies macros (like applying a mod which change the analysis 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 created

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



AWS Solutions Architect Associate

CloudFormation



AWS Quick Starts



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.

Architecture Diagram:

[Switch to full-screen view](#)



AWS Solutions Architect Associate

CloudFormation



CloudFormation Cheat Sheet



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 macros (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



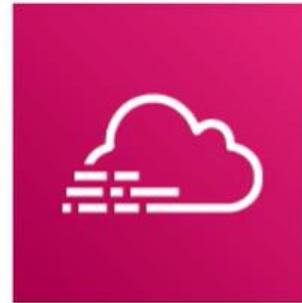
AWS Solutions Architect Associate

CloudTrail



CloudTrail Introduction

CloudTrail



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



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

```
1 {"Records": [
2     "eventVersion": "1.0",
3     "userIdentity": {
4         "type": "IAMUser",
5         "principalId": "EX_PRINCIPAL_ID",
6         "arn": "arn:aws:iam::123456789012:user/Worf",
7         "accountId": "123456789012",
8         "accessKeyId": "EXAMPLE_KEY_ID",
9         "userName": "Worf"
10    },
11    "eventTime": "2014-03-24T21:11:59Z",
12    "eventSource": "iam.amazonaws.com",
13    "eventName": "CreateUser",
14    "awsRegion": "us-east-1",
15    "sourceIPAddress": "127.0.0.1",
16    "userAgent": "aws-cli/1.3.2 Python/2.7.5 Windows/10",
17    "requestParameters": {"userName": "LaForge"},
18    "responseElements": {"user": {
19        "createDate": "Mar 24, 2014 9:11:59 PM",
20        "userName": "LaForge",
21        "arn": "arn:aws:iam::123456789012:user/LaForge",
22        "path": "/",
23        "userId": "EXAMPLEUSERID"
24    }}
25 ]}]
```



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.**

CloudTrail

Dashboard

Event history

Trails

Learn more

Pricing ↗

Documentation ↗

Forums ↗

FAQs ↗

Event history

Your event history contains the activities taken by people, groups, or AWS services in supported services. It filters out read-only events. You can change or remove that filter, or apply other filters.

You can view the last 90 days of events. Choose an event to view more information about it. To view a complete trail and then go to your Amazon S3 bucket or CloudWatch Logs. Learn more

Can't find what you're looking for? Run advanced queries in Amazon Athena

Event time	User name	Event name
2019-09-01, 09:33:07 PM	i-014d0d0e482491e69	UpdateInstanceInformation
2019-09-01, 09:30:07 PM	i-08ece9e263d3edfbc	UpdateInstanceInformation
2019-09-01, 09:28:07 PM	i-0984241e0f6a0f9ca	UpdateInstanceInformation
2019-09-01, 09:25:07 PM	i-07a9e824ebb4d5f2b	UpdateInstanceInformation
2019-09-01, 09:23:34 PM	exampro-events	CreateLogStream
2019-09-01, 09:23:07 PM	i-014d0d0e482491e69	UpdateInstanceInformation
2019-09-01, 09:20:07 PM	i-0f5f9d47f3c1cf6d	UpdateInstanceInformation
2019-09-01, 09:18:07 PM	i-08ece9e263d3edfbc	UpdateInstanceInformation
2019-09-01, 09:15:07 PM	i-07a9e824ebb4d5f2b	UpdateInstanceInformation
2019-09-01, 09:13:51 PM	exampro-metrics	CreateLogStream



AWS Solutions Architect Associate

CloudTrail



CloudTrail Trail Options



CloudTrail - Trail Options

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

Apply trail to all regions Yes No

Creates the same trail in all regions and delivers log files for all regions

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

Apply trail to my organization Yes No 

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

Encrypt log files with SSE-KMS Yes No 

Create a new KMS key Yes No

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

Create a new S3 bucket Yes No

S3 bucket* 

[Advanced](#)

Log file prefix 

Location: /AWSLogs/655604346524/CloudTrail/us-east-1

Encrypt log files with SSE-KMS Yes No 

Enable log file validation Yes No 

Send SNS notification for every log file delivery Yes No 



AWS Solutions Architect Associate

CloudTrail



CloudTrail to CloudWatch



CloudTrail to CloudWatch

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

▼ CloudWatch Logs

Configuring delivery to CloudWatch Logs enables you to receive SNS notifications from CloudWatch when specific API activity occurs. Standard CloudWatch and CloudWatch Logs charges will apply. [Learn more](#)

Configure



AWS Solutions Architect Associate

CloudTrail



Management vs Data Events



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

▼ Data events

Data events are logs of resource operations performed on or within a resource.

S3

Lambda



AWS Solutions Architect Associate

CloudTrail



CloudTrail Cheat Sheet



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 90 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